Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

"History of MPEG Video Compression" – Ver. 4.0

Comments on this version:

This document is a draft of the final version, which will be completed in a few months time. This version contains a detailed description of the technology through the 1950s, 1960s and 1970s. It includes a detailed description of the H.261 standard development. The description is however basically a recitation of the facts with only some analysis and little conclusion. This version includes a reasonably detailed description of the technology in the 1980s, and an overview of the technology in the 1990s.

In the final version of the document, the same level of detail will be provided for the 1980s and 1990s, including the MPEG1, MPEG2, MPEG4 and H.263 standards. The document will include more analysis of the work, and in particular will provide summaries and conclusions that can be drawn from all the experience gained over the years from implementing the technology and continuously enhancing it.

The work is believed to be accurate, but I welcome any comments concerning the accuracy and completeness of the text.

Cliff Reader.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

Introduction

The MPEG (Motion Picture Experts Group) video-coding standard is not a single algorithm but rather an integrated set of coding tools. The combination of these tools leads to the total compression ratio. Although MPEG has only recently become very popular, the history of video compression goes back a long way. The main tools that provide most of the performance were developed over thirty years ago, and the basic MPEG-style algorithm existed in 1981. Originally each of the main tools was investigated as a complete solution for video coding and to an extent they competed against each other. The history can thus be viewed as threads of development, originally in parallel, but converging to an integrated solution whose performance exceeds any one of them. The CCITT SGXV and ISO WG11 committees[1] were forums for this convergence, where experts from many companies and countries pooled their efforts to yield this superior performance. In specific cases, solutions were even invented by and within the committees, particularly in recent years. Although the technology was invented many years ago, it found little practical application at that time because it was too expensive to implement. Only since the mid-1990s has semiconductor technology advanced to the point where real-time video processing can be provided for consumer applications. Early use was limited to military, medical, distance learning and videoconferencing applications. Many of the early developments in image and video processing were forgotten, only to be reinvented in the 1990s by a new generation of researchers. This paper will document the original developments and show both how they became integrated into the MPEG standard and what their relative contribution is to the overall performance.

The next section of the paper provides an overview of video compression, and introduces the main MPEG coding tools. The overall timeline is shown for invention and evolution of these tools and the development of the major video coding standards. The following section then explains the foundation for video coding from the construction of the analog video signal to the early attempts to perform compression in the analog domain. The history of early digital coding is given, and together these topics comprise the history during the 1950s and 1960s. The early work on statistical characterization of digital video signals is described, followed by introduction of the models that were developed for the analysis of digital video. The section concludes with a discussion of the degree to which the measured statistics and mathematical models accurately describe the data, and the extent to which they can be used to design coding systems. Having set the stage, the paper then includes two substantial sections describing the development of the major tools for spatial compression and temporal compression. Each one shows a timeline for the various coding tools and discusses the development of each tool. The third major tool, statistical coding is covered within these sections. Comprehensive references provide the first published paper for each piece of coding technology, and papers describing the

[1] The international committees evolve over time to reflect current requirements for their services and expertise. The CCITT is now ITU-T, and the committee responsible for video coding is now SG 16. At the time of founding, MPEG was an ad hoc group within ISO/IEC JTC1/SC2 WG8. It became an independent working group as ISO/IEC JTC1/SC29 WG11.

milestones of evolution of the technology to the final form in which it is used by the standard. The paper then describes the development of the coding standards, in particular H.261. This standard was the model for all the ones coming after it like MPEG1 and MPEG2. Development of the standards forced the coding algorithms to be augmented with all the features necessary for complete working solutions, which resulted in the development of numerous small coding tools. The next section of the paper classifies and describes those tools. The final section consists of an evaluation of the performance of each tool and its relative contribution to the overall performance.

Overview

Video compression is possible and necessary because raw video data is highly redundant. This is true of analog video, which will be discussed briefly later, but it is particularly true of digital video. Compression concerns the removal of redundancy and is based on two techniques – statistical and psychovisual. The opportunity to remove statistical redundancy is created by the regular sampling structure of digital video in both the spatial and temporal dimensions. Video frames are digitized into regular arrays of pixels with a density suitable for representing the highest spatial frequency at every point. Naturally, most frames contain very little or no data of such maximum detail. Similarly, the frame rate of video is chosen to represent the maximum motion in a scene, and again such motion occurs infrequently, and even then usually only in part of the frame. Ideally, a compression system would provide only the instantaneous information necessary to describe the scene. Obviously this would need to be highly adaptive to the data in the scene, and the data rate so produced would vary dynamically. The opportunity to use psychovisual techniques is created by the limitations of the human vision system. The vision system has limitations of contrast bandwidth, spatial bandwidth (especially for color vision), and temporal bandwidth. Notably, these limitations are not independent, and there is effectively a limit to overall visual processing bandwidth such that, for example, it is not possible to simultaneously perceive high spatial and temporal resolution. Clearly it is pointless to present information that cannot be perceived, and compression schemes take advantage of that fact.

Three major coding tools provide most of the performance of MPEG. Adaptive block transform coding provides removal of spatial redundancy. Motion-compensated DPCM provides removal of temporal redundancy. These two tools are combined in a technique known as hybrid coding. Entropy coding is then used to remove the statistical redundancy in the codes produced by the hybrid coder. These major tools are augmented by a number of minor coding tools that remove residual redundancy in particular portions of the coded data, adapt the coding to specific applications, or provide the means to format the data into a bitstream for storage or transmission.

This paper is organized around the threads of technology that introduced, refined, and merged all these various coding techniques. Each major tool is traced from its inception through to incorporation in the standard in its final form. Minor tools are shown as contributors along the way. In some cases dead ends are shown where a particular

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

technology yielded to another. The threads are illustrated diagrammatically with timelines, and at key points along the development path, figures show block diagrams of encoder or decoder configurations. An assessment is made of the relative contribution of each tool to the total compression performance. This is expressed in terms of bits/pixel.

The overall timeline is illustrated in Figure 1. This shows the period of development for the three major tools and for the International Standards developed by the ITU and ISO/IEC. The starting date for each tool is the date on which the first publication was made and when it was first applied to video. The ending date shown is the approximate date on which substantive development that relates to the standards was complete. The dates for the standards are the ones for inauguration and substantial completion.

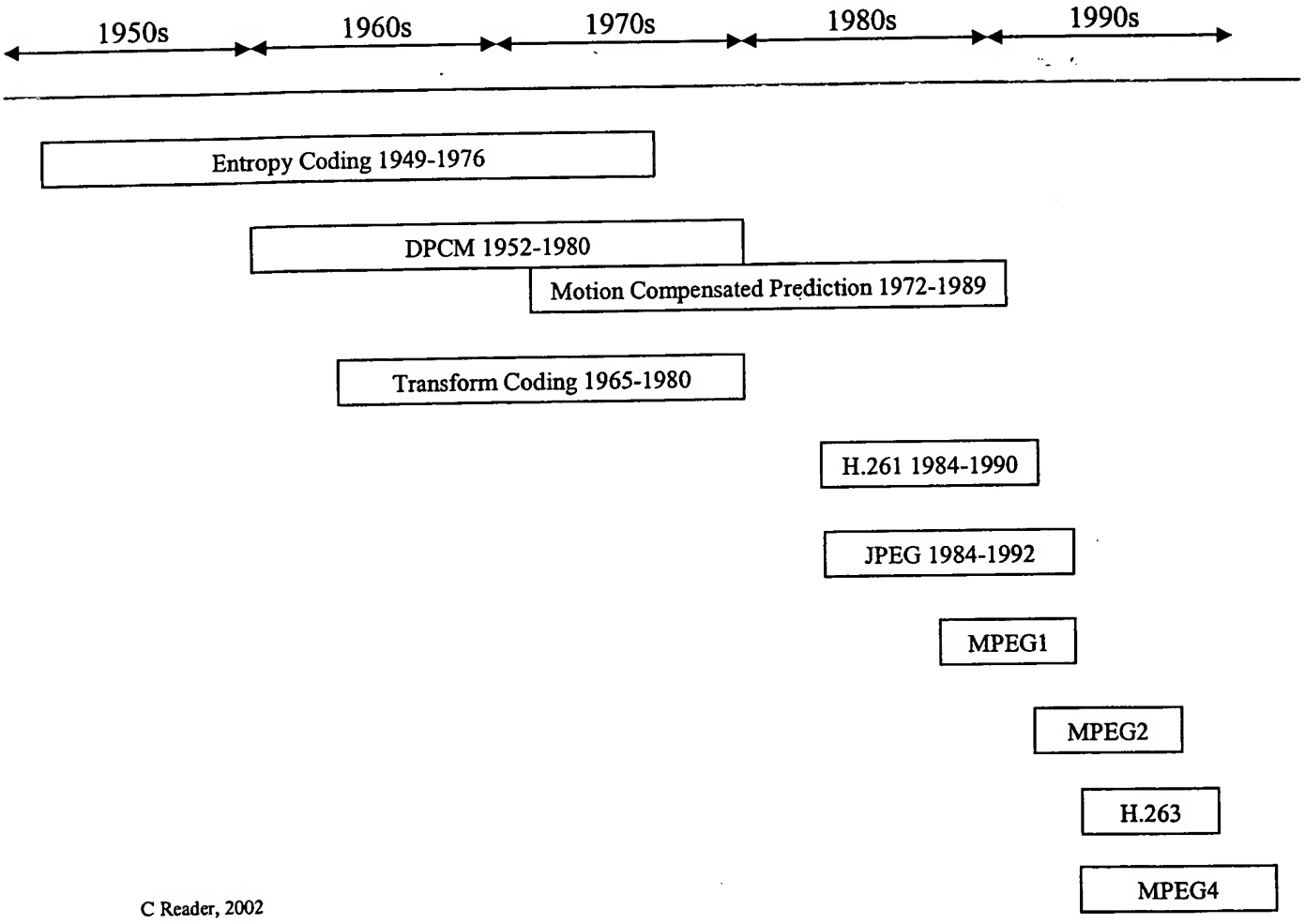


Figure 1. Development of Coding Tools and Standards

The modern form of entropy coding was invented in the late 1940s[2]. It was applied to video coding in the late 1960s, and evolved until the introduction of 2D VLC coding and arithmetic coding in the mid 1980s. DPCM[3] was invented in 1952, and was first applied to video in the same year. It was initially developed as a spatial coding technique only, but by the mid 1970s DPCM was also being used in the temporal domain. DPCM evolved until the early 1980s, at which point it ceased to provide a standalone solution to video coding. Starting in the early-mid 1970s, key elements of DPCM coding were merged with Transform coding to create Hybrid coding. This evolved into the form used in MPEG by the early 1980s.

Transform coding was first applied to video in the late 1960s. Substantial development during the first half of the 1970s established transform coding as the highest performance solution for spatial coding. In the Hybrid coder, transform coding is used to remove spatial redundancy, while DPCM is used to remove temporal redundancy. The performance of temporal DPCM was greatly improved by motion compensated prediction. This was invented in 1969, and evolved into the basic form used by MPEG by the beginning of the 1980s. In the early 1980s it was extended to interpolative coding in which prediction was made across multiple frame intervals and the intermediate frames were predicted with scaled motion vectors. It did not reach its final form however until the end of the 1980s when bi-directional prediction was introduced[4]. In recent developments (H.26L), prediction quality has been improved. This means that the difference signal is much less correlated. As a result the transform is less required, and H.26L uses a reduced transform (4x4).

The CCITT H.261 standard was started in 1984, and was substantially complete by 1989[5]. This standard is a direct precedent for MPEG. MPEG1 and H.261 share common data structures, coding tools and syntactic elements. While the two are not precisely backward compatible, MPEG1 can be viewed as a superset of H.261. MPEG1 development began in 1988, and was substantially complete by 1992. MPEG2, which is a superset of MPEG1, began in 1990, and was substantially complete in 1994. H.263 started in 1992 and version 1was completed in 1995. MPEG4 (where the video was built on MPEG2 and H.263) was started in 1993, and version 1 was substantially complete in 1998.

In addition to compression technology itself, there are two other issues of importance discussed in the paper. First, the data structure has a fundamental impact on the design of the end-to-end video system, and second, the statistics of the data strongly influence the coding algorithm design. For the former, it is important to realize that digital video is evolutionary, not revolutionary, in that the fundamental data structure of analog video is

[2] However, earlier examples exist. Morse code, invented in the 19th century is an obvious example.
[3] "DPCM" is frequently used as a term for predictive coding.
[4] It might be argued that it is still evolving with the recent developments originating in ITU H.26L work.
[5] The International Standards all have an extensive period of ratification between substantive technical completion and formal publication.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

unchanged. MPEG encodes a sequence of two-dimensional frames. This constrains the coding algorithm to operate on the spatial information present when an image of the three-dimensional world is projected onto the two-dimensional plane of the video camera, and to operate on the temporal information provided by a regular sequence of frames. The loss of truly three-dimensional information, and the discretization of motion have significantly influenced the algorithm design. Second, the spatial and temporal statistics of video data play a key role in the algorithm design. In particular it will be seen that the statistics are not stationary, which leads to locally adaptive and multi-modal algorithm design.

The MPEG algorithm has been fully described elsewhere and is not the subject of this paper. The encoder and decoder are illustrated in Figure 2.

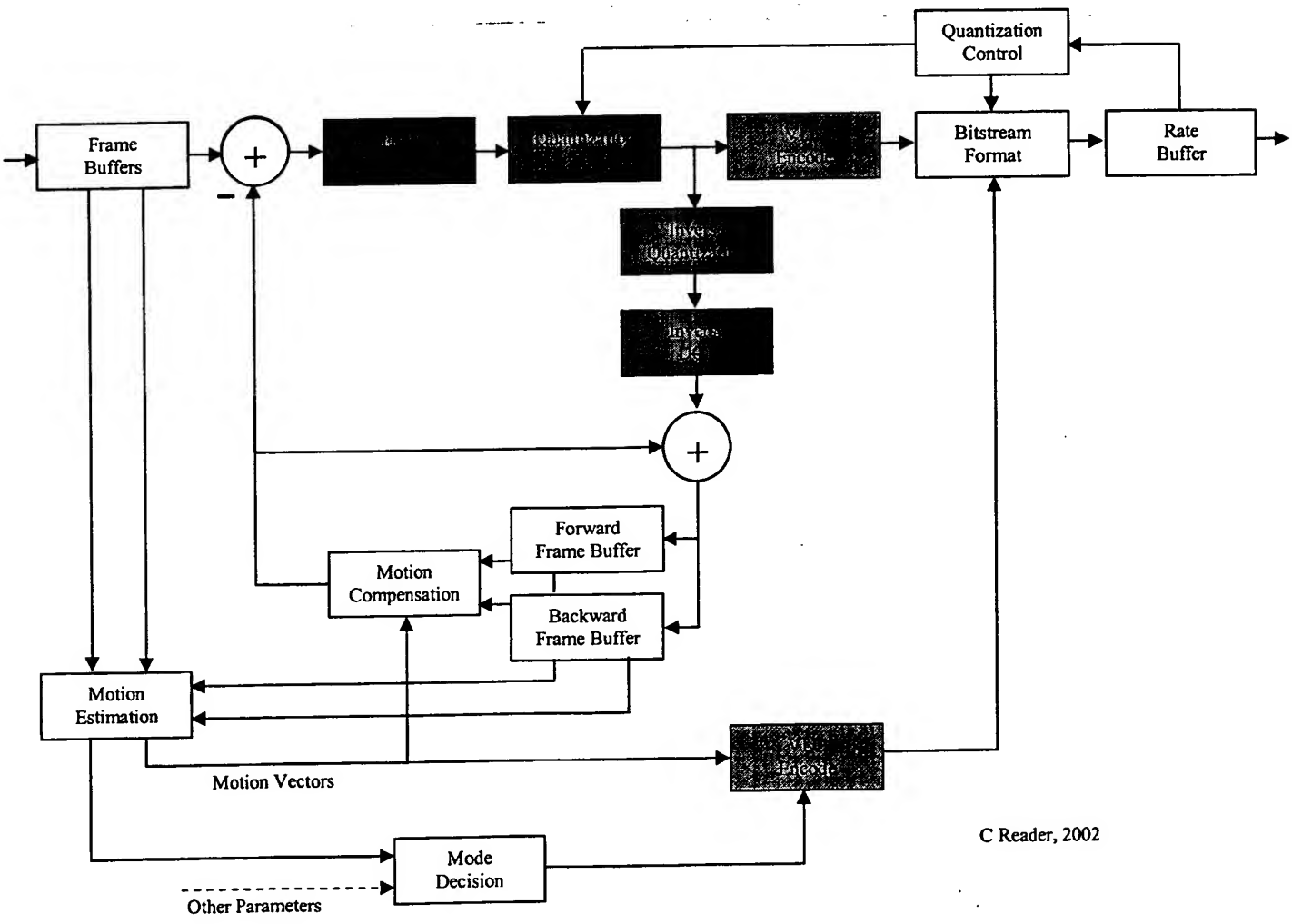


Figure 2a. MPEG Video Encoder

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

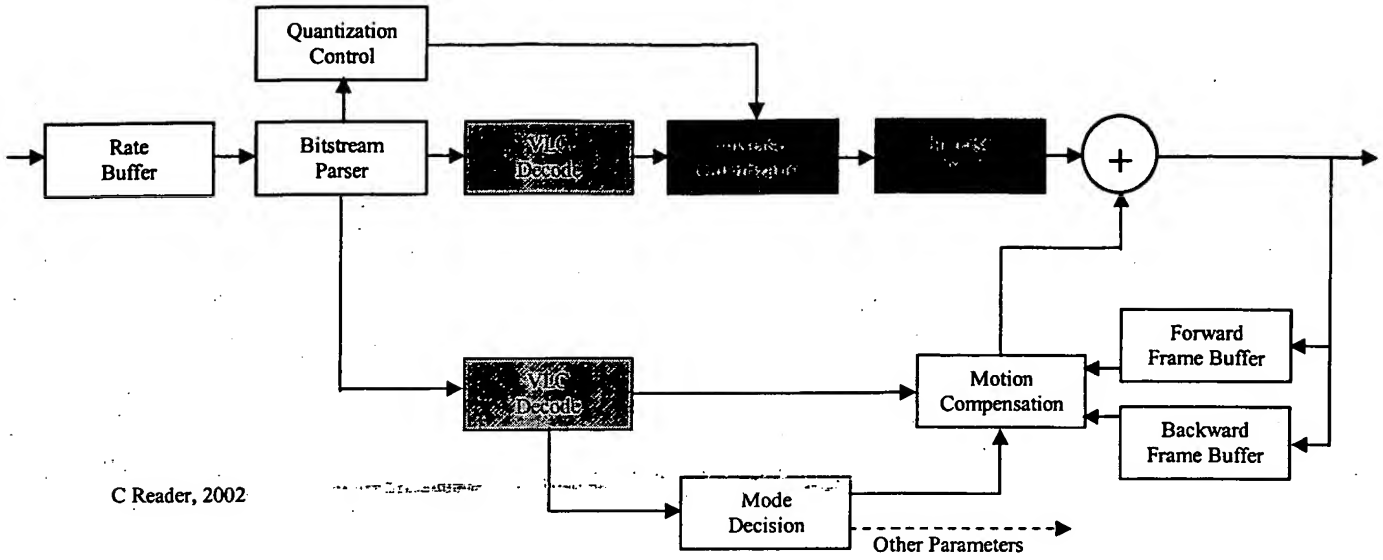


Figure 2b MPEG Video Decoder

The coding tools used in the MPEG1/MPEG2 algorithm are listed in Table 1, grouped according to the major tools. Table 2 contains additional normative requirements for the MPEG syntax and decoding process. The tables cover all of the normative requirements of the MPEG1 standard in the syntax and the most significant normative requirements of the MPEG2 standard. The layer of the syntax that signals the tool is shown in parenthesis. A few normative requirements are not signaled and are known a priori by decoders.

There have been many surveys, bibliographies and conferences on video compression. References [] to [] provide a comprehensive view of the state-of-the-art at intervals spanning the development of this technology.

- Cherry 1962;
- Proc. IEEE Mar 1967.
- Wilkins and Wintz, IT-19, Mar 1971.
- Proc. IEEE Jul 72
    - Wintz.
    - Haskell et al.
- Haskell chapter in Pratt book, Image Transmission Techniques.
- Netravali, Limb, Proc. IEEE Vol. 68, Mar 1980
- Jain, Proc. IEEE, Vol. 69, Mar 1981.
- Motion compensation overview paper by Ishiguro 1982
- Proc IEEE Vol. 73, No. 4, Apr 1985
    - Mussman – survey – both pixel and transform
    - Kretz/Nasse – survey at system level – both again
- Srinivasin, SPIE Vol. 528, 1985 – Teleconferencing survey paper.
- Rao, SPIE Vol. 757, 1987 – summary paper for all the different approaches.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

Temporal coding	Motion Compensated Prediction Prediction modes (Picture, MB) MPEG1 progressive modes MPEG2 progressive and interlace modes Out-of-order decoding (Picture) MV Range (Picture, MB) Integer & ½-pixel (MPEG2 ½-pixel only) (Picture) MV prediction (MB)
Spatial coding	Adaptive block DCT Quantizer Uniform quantizer scale Deadzone in interframe Scale factor (Slice, MB) Downloadable quant. Matrices (Sequence) Intra/non-intra Luma/Chroma (MPEG2 only) Alternate quantizer scale (MPEG2 only) Scan pattern ZZ-scan Alternate scan (MPEG2 only) (Picture) Intra DC-term size (MPEG2 only) (Picture) Intra DC-term prediction (Block)
Entropy Coding	Coefficient VLC coding 1st coeff. (Block) Escape code (MPEG1 double escape code) (Block) EOB (Block) Intra VLC format (MPEG2 only) (MB) Motion vector coding MB Address; Skipped MB; Escape code (MB) MB Type VLC (MB) Prediction mode Forward prediction flag Backward flag (Both forward and backward – bi-directional) (Neither forward nor backward – intra) Scale factor flag (Mquant) CBP (Coded/Not-coded)

Table 1. MPEG Coding Tools

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

IDCT Mismatch	Oddification (MPEG1 only) Coeff [7,7] (MPEG2 only) Forced update (MB)
Rounding & Clipping	IDCT "/" operator in inverse quantization "//" operator in motion compensated prediction
Bitstream Formatting	Start codes (All layers) Closed GOP/Broken Link (GOP) Slice Vertical Position (Slice)
Decoder initialization	Picture size, rate, aspect ratio (Sequence) Bitrate (Sequence) Buffer size (Sequence) Chroma format (MPEG2 only) (Sequence)
Rate Buffer	Buffer size[6] (Sequence) Buffer delay (Picture) Stuffing Macroblock (MPEG1 only) Zeros before start codes
Sync, Error Resilience	Start codes (Sequence, GOB, Picture, Slice) Slice vertical position (Slice) Temporal reference (Picture) Intra Concealment motion vectors (MB)

Table 2. Additional MPEG Normative Requirements[7]

The Foundation of MPEG Coding

The origins of video compression can be found in the earliest days of television itself. From the beginning the system was designed to accommodate and exploit the limitations of the human vision system. Viewed from this perspective, interlace is a 2:1 compression algorithm that is based on the different perception of human vision for motion and flicker. It was already known from film that the perception of smooth motion could be achieved with around 24 frames per second but that flicker could be perceived at up to about twice that rate. In the movie theater, displaying each frame twice solved the problem. Such a solution for television would have required a frame store in the receiver – an impractical idea until very recently. Then when color television was introduced, the limited response of the human vision system to high spatial frequencies in color, led to YIQ coding in the NTSC color television system. The primary colors are transformed into luminance-chrominance space, and the chrominance signals are filtered and allocated

[6] MPEG1 Constrained Parameter Bitstreams & MPEG2 Main Profile Bitstreams.
[7] Table 1 and Table 2 do not include the following differences between MPEG1 and 2: chroma sample location, slice formats, D-pictures, aspect ratio definition, MPEG1 constrained parameter bitstream vs. MPEG2 profiles and levels, variable bitrate and vbv_delay definition, temporal reference definition, ancillary MPEG2 information downloads such as composite video.

only a fraction of the bandwidth of the luminance signal. In the NTSC system, the transformation of the two chrominance signals is such that they correspond to orthogonal axes of maximum and minimum perceptual discrimination of color difference. Therefore the analog color TV system compresses three primary-color signals each with approximately 4MHz bandwidth into a luminance signal with 4MHz bandwidth and two chrominance signals with 1.3MHz and 0.5MHz bandwidth. These signals are modulated into a single 5MHz signal. In the early 1980s, an equivalent digital standard was set for studio-quality video. Originally designated CCIR 601, now ITU-R BT.601, this standard filters and subsamples the chrominance signals horizontally by a factor of two.

Prior to the introduction of digital video, there were a number of efforts to provide additional compression in the analog domain. The earliest attempt was made by Kell [1], who in 1929 patented a technique for conditional updating. It is not clear whether he implemented the technique, which comprised sending only the areas of frames that had changed since the previous frame. The scheme used the inductance of transformer coils as the equivalent of a frame store. Conditional updating is a direct ancestor of MPEG coding. Other analog schemes are not relevant to MPEG, and will not be discussed further here.

In 1949[8], Goodall [2] first presented the idea of applying pulse code modulation – PCM – to television. The work described a 10MHz sampling rate and varied the number of bits in a uniform quantizer up to 5 bits. Contouring was shown when too few bits were used, and the masking effect of additive noise was presented. This paper was followed by a number of papers exploring the statistics of digitized video and developing new coding techniques. Kretzmer 1952 [3] examined the probability distribution of pixel amplitudes, the probability distribution of a first order linear predictor, and the autocorrelation function. His discoveries for the linear predictor set the stage for the work on DPCM coding. This will be discussed in a later section of the paper. However, he noted that such an approach ignored "the bulk of the large-scale redundancy, which is of a far-flung and nonlinear nature". In this he anticipated the highly adaptive nature of the eventual high-performance MPEG algorithm, and the difficulty of using analytical tools to describe it. Kretzmer also anticipated the need for 8 or 9-bit PCM for high-quality video and the use of Shannon-Fano coding to remove the redundancy of first-order predicted data. A series of papers followed that examined the use of non-uniform quantization and addition of pseudo-noise for PCM coding. See for example, Kretzmer 1956 [4], Oliver 1952 [5], Roberts 1962 [6], Graham 1962[7] and Thomson 1967 [8]. The subjective effects of video noise were studied Brainard et al 1962 [9]. One finding of the studies was the psychovisual effect that quantization error is less visible in areas of high spatial frequency. This led to development of a coding scheme first proposed by Kretzmer 1956 [4], in which the high frequencies were more coarsely quantized. It was extensively developed by Schreiber et al 1959 [10] into a technique known as synthetic highs. The technique of quantizing the high frequencies more coarsely is reflected in the MPEG standard in the default matrices for intra-coded pictures that have subjectively weighted

[8] The Paper was orally presented in 1949 and published in 1951.

quantization matrices. Statistical analysis of video was also performed by Seyler 1962 [11] 1963 [12] 1965 [13], Seyler and Budrikis 1965 [14], Schreiber 1956 [15], and Franks 1966 [16]. Seyler's paper [11] provides a good overview of the work up to that point (1962). This paper also discusses the properties of the human vision system, and discusses the interdependence of contrast resolution, spatial resolution and temporal resolution. It is as though the human vision system contains some total bandwidth that at any given instant may be allocated adaptively to more contrast resolution (in flat areas of images), or spatial resolution (in areas of high detail) or temporal resolution (in areas of fast motion). Seyler and Budrikis' work also covered the important finding that the vision system cannot perceive full detail for approximately ¾ seconds after a scene change. This allows coding schemes such as MPEG to present coarser quantization after a scene change. Entropy coding was introduced in 1949 by Fano [17] with the so-called Shannon-Fano code, followed by Huffman [18] who defined the set of minimum redundancy codes.

The early work on statistical analysis and experimentation was followed a period of formal definition. A simple model for statistical coding was introduced by Huang and Schultheiss [19] who invented the technique of block quantization. The model comprises a pre-processing operation to segment the data into blocks, followed by a two-step quantization process. This is illustrated in Figure 3.

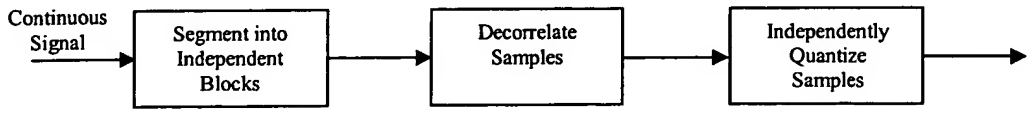


Figure 3. Block Quantization.

First, the data are segmented into blocks – these are often two-dimensional in image or video coding. These blocks are assumed to be statistically independent, if all of the redundancy is to be removed. Second, the data samples are decorrelated. Third, the decorrelated samples are independently quantized using a model of their statistics or empirically defined statistics. Both transform coding and predictive coding follow this basic approach, differing basically in the method for decorrelation.

The statistical models developed in the early years of video compression apply to a large ensemble of frames. While such models apply to most areas of most video frames, the global statistical parameters may not be accurate for local areas of individual frames. In other words the statistics are not stationary. To deal with this, the models should be parameterized so they can be made locally adaptive. However an additional problem concerns local discontinuities such as high-contrast edges or scene changes, for which the model will not apply at all. Therefore applying a coding algorithm based on such a model will fail at that location, perhaps completely. Such non-linearity makes mathematical analysis very difficult. Worse yet, such local discontinuities contain the very information of most human interest, and the eye will be drawn to any coding artifact at such a location. This perceptual error will not correlate well with analytical measures such as mean square error (MSE), because the analytical measures are based on ensemble

statistics and the majority of well-behaved areas of the frame will overwhelm the small areas of local discontinuity. Throughout this paper there will be frequent discussion of this point. In many ways the success of MPEG is due to its flexibility in managing non-stationarity parametrically, and providing fallback strategies when the coding algorithm fails, that minimize both visibility of errors and artifacts and the loss of compression efficiency at that point.

Section I References

1. R D Kell, British Patent No. 341,811, 1929.

2. W M Goodall, "Television by Pulse Code Modulation", BSTJ Vol. 30, pp. 33-49, 1951.

3. E R Kretzmer, "Statistics of Television Signals", BSTJ, Vol. 31, No. 4, pp. 751-763, Jul 1952.

4. E R Kretzmer, "Reduced-Alphabet Representation of Television Signals", IRE Convention Rec., Pt. 4, pp. 140-147, 1956.

5. B M Oliver, "Efficient Coding", BSTJ pp.724-750, Jul 1952.

6. L G Roberts, "Picture Coding Using Pseudo-Random Noise", IRE Trans. On Information Theory, Vol. IT-8, No. 2, pp. 145-154, Feb 1962.

7. R E Graham, "Snow Removal – A Noise-Stripping Process for Picture Signals", IRE Trans. On Information Theory, Vol. IT-8, No. 2, pp. 129-144, Feb 1962.

8. J E Thomson, J J Sparkes, "A Pseudo-Random Quantizer for Television Signals", Proc. IEEE, Vol. 55, No. 3, pp. 353-355, Mar 1967.

9. R C Brainard, F W Kammerer, E G Kimme, "Estimation of the Subjective Effects of Noise in Low-Resolution Television Systems", IRE Trans. On Information Theory, Vol. IT-8, No. 2, pp. 99-106, Feb 1962.

10. W F Schreiber, C F Knapp, N D Kay, "Synthetic Highs – An Experimental TV Bandwidth Reduction System", SMPTE Journal, Vol. 68, pp. 525-537, Aug 1959.

11. A J Seyler, "The Coding of Visual Signals to Reduce Channel Capacity Requirements", IEE Monograph No. 535 E, pp. 676-684, Jul 1962.

12. A J Seyler, "Real-Time Recording of Television Frame Difference Areas", Proc. IEEE, pp. 478-480, Mar 1963.

13. A J Seyler, "Statistics of Television Frame Differences", Proc. IEEE, pp. 2127-2128, Dec 1965.

14. A J Seyler, Z L Budrikis, "Detail Perception after Scene Changes in Television Image Presentations", IEEE Trans on Info. Theory, pp.31-43, Jan 1965.

15. W F Schreiber, "The Measurement of Third-Order Probability Distributions of Television Systems", IRE Trans. on Info. Theory, Vol. IT-2, pp. 94-105, Sep. 1956.

16. L E Franks, "A Model for the Random Video Process", BSTJ, pp. 609-630, Apr 1966.

17. R Fano, "Transmission of Information", Research Lab. Electronics, MIT, Tech. Rep. 65, 1949. Also, MIT Press and Wiley pp.170-172, 1961.

18. D A Huffman, "A Method for the Construction of Minimum Redundancy Codes", Proc. IRE, Vol. 40, No. 10, pp.1098-1101, 1952.

19. J Y Huang, P M Schultheiss, "Block Quantization of Correlated Gaussian Random Variables", IEEE Trans. On Comm. Sys., pp. 289-296, Sep 1963.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

Section II Spatial Coding

Adaptive Block Transform Coding

It was observed earlier that frames of digital video are sampled with a uniform array of samples, with sample spacing capable of reproducing the highest detail at every point. This can also be expressed in terms of spatial frequency in a manner directly equivalent to the frequency representation of time-varying signals, but extended to two dimensions. Then the image may be analyzed or processed by use of Fourier techniques. In natural scenes, there will be a predominance of low-frequency information. So while in the spatial domain, the signal energy is uniformly distributed among the samples, in the transform domain, the energy is concentrated in a few samples. This observation led to the development of transform coding in the latter half of the 1960s. The timeline of development for transforming image data, and for coding it efficiently are shown in Figure 4.

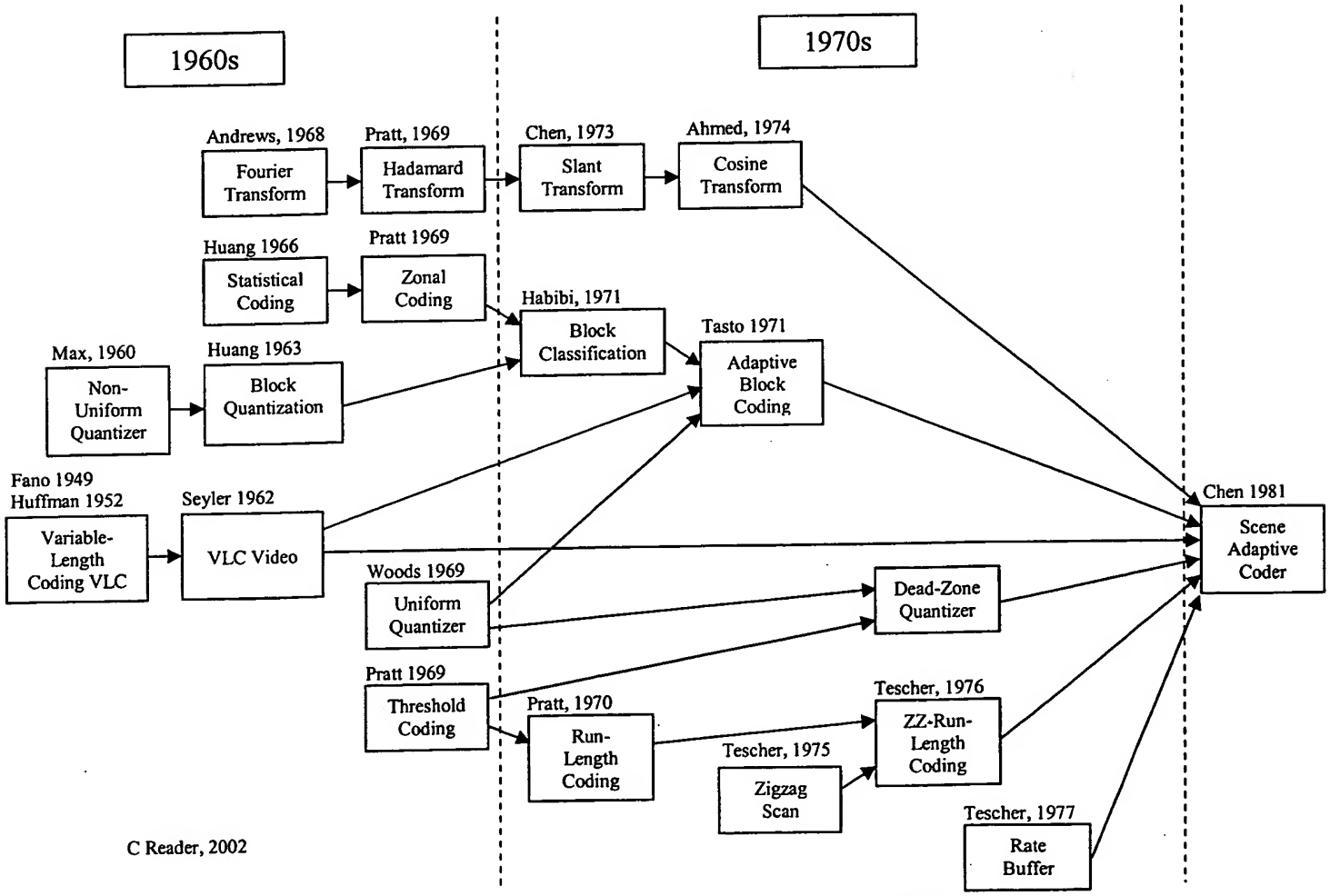


Figure 4. Transform Coding Timeline (to 1981)

Five threads of development are illustrated in this figure:

- The transform
- Zonal coding
- Threshold coding
- VLC (entropy) coding
- Rate buffering

These threads converged to the design of the first fully implemented transform coder at the end of the 1970s. Two techniques became dead-ends: zonal coding, and the Max non-uniform quantizer.

The Transform

The first work was reported by Enomoto and Shibata [II-1] in 1965, which studied the use of small (length eight), one-dimensional Hadamard transforms and a block quantization scheme. Andrews and Pratt [II-2] first studied the use of two-dimensional transforms in 1968. Initially, the coding was performed by transforming the entire image with a single transform. (A typically size was 512x512 pixels). It was quickly realized that this approach was both very expensive to implement and unnecessary. The number of computations to transform a vector of length N using a "fast" transform[9] is proportional to $N\log_2 N$, while correlation between pixels falls to a low level for pixels spaced approximately sixteen pixels apart[10]. Therefore pictures were divided into blocks that were transformed and coded independently. Andrews and Pratt used the Fourier transform initially, however two limitations of this transform were also quickly realized. Computation of the transform required high-precision complex arithmetic involving multiplication. (Usually implemented in floating-point computation). This made any kind of real-time implementation unfeasible at that time. Also, the periodic structure of the Fourier transform meant that when it was used to transform a finite-length vector, the resulting spectrum contained frequencies representing the transition between the pixels at opposite ends of the vector. Since these pixels were largely uncorrelated, there was usually a significant brightness transition between them that produced high-frequency energy in the Fourier domain. It was either inefficient to code this energy, or if it was ignored, the edges of the block would be attenuated (effectively because a window function had been applied), and the block structure became visible as an artifact. This led to investigation of other transformations.

The theoretically optimum transform[11] is composed of basis functions that are the eigenvectors of the matrix that diagonalizes the covariance matrix of the data. This is the Karhunen-Loeve (K-L) transform and is also known as Hoetelling's method of principal components. See Habibi and Wintz 1969 [II-3] 1971 [II-4]. This transform completely decorrelates the data, assuming however that the statistical model is accurate. As

[9] For a direct transform the number of computations is proportional to N^2.
[10] An exception is the DC term which is often correlated over much larger pixel spans. Coding the DC term is discussed later.
[11] In a mean-square-error sense.

discussed earlier, statistical models are only approximate therefore the practical performance is less than ideal. For this reason and because of computational complexity[12], the K-L transform is not used in practice[13].

For some years, attention was focused on the Hadamard Transform, also known as the Walsh Transform. Pratt, Kane and Andrews in 1969 [II-5] provided a thorough analysis, and in the same year, it was also examined by Woods and Huang [II-6] together with block quantization. The Hadamard transform attracted attention because it could be computed using only addition and subtraction. (At that time, multiplication was very expensive[14]). It was shown theoretically that the Hadamard transform provided more energy compaction than the Fourier transform for small block sizes, and it did not suffer from the problem of visible block boundaries. However, the perceptual error produced by the Hadamard Transform was a problem, and this illustrates the difference between analytical measures of performance such as MSE (mean square error) calculations and perceived error. The Hadamard basis functions are square-wave in nature, and do not correspond to the natural structure of real-world scenes. As a result, truncation of the Hadamard series and quantization of Hadamard coefficients for coding purposes causes visibility of the basis functions, and therefore a blockiness artifact. Notably, diagonal edges become rendered as staircases. Two other transforms were investigated. The Haar transform possesses even simpler computation, but its performance was low[15]. The Slant transform was invented as a handcrafted modification of the Hadamard transform in which some of the low-frequency basis functions were replaced by ramp-like functions that were more natural in appearance 1973 [II-7].

These efforts were completely supplanted in 1974 by the invention of the Cosine transform by Ahmed, Natarajan and Rao [II-8]. The theoretical performance of the cosine transform approaches that of the K-L transform for highly correlated data, and it has natural sine-wave basis functions that tend not to produce visible artifacts. A simple way to understand the Cosine transform is its derivation from the Fourier transform. Due to its periodicity property the Fourier transform of a symmetric sequence is real – it only has cosine terms. If a finite sequence of samples is duplicated by reflecting it about one end point, the resultant sequence is symmetric. If this is Fourier transformed, the coefficients are all real, and in fact the computation can be formulated such that only real arithmetic is necessary. In addition, there are no spurious high-frequency terms because there are no discontinuities at the ends of the reflected sequence of samples. The cosine transform elegantly solves both the problems with the Fourier transform. From this point on, the

[12] The basis functions for the transform itself must be computed before being applied to the data. The computation of the basis functions must be performed every time the local statistics change, and the computation is very intensive involving matrix inversion.
[13] The K-L transform has found application for multispectral data processing where it can be used in the spectral dimension to decorrelate the small number of multispectral bands found in typical applications.
[14] The first VLSI multiplier chip was not introduced until 1974 (TRW MPY8A). The first VLSI multiplier-accumulator chip was introduced in 1975 (TRW TDC1003) and the first FIR filter chip in 1981 (TRW TMC2242).
[15] The Haar transform is related to the more recent wavelet technology.

Cosine transform was used almost exclusively, and considerable effort was devoted to finding fast algorithms to minimize the computational burden. Such algorithms optimized the calculations to satisfy a number of tradeoffs such as minimizing the number of multiplications, or providing the smoothest data flow with minimum control. Other features have also been supported, notably the hierarchical feature in which the transform of some maximum size is implemented as a cascade of smaller-sized transforms. Such an implementation can be used in adaptive algorithms varying the transform size parametrically.

Coding Strategy

The transform produces a matrix of almost decorrelated coefficients. For natural scenes, the energy or variance of the samples is non-uniformly distributed. Typically, there is a concentration of energy in the low-frequency coefficients. These samples should be allocated a reasonable number of bits for coding and quantized reasonably finely. The higher frequency samples can be allocated fewer bits and quantized coarsely. Many of the highest frequency samples need not be coded at all. The coarse coding of high-frequency samples is appropriate both in terms of their statistical significance and in terms of their subjective importance [I-4][I-10]. Huang and Shultheiss 1963 [I-19] showed that after the samples were decorrelated by the transform process, they should be assigned coding bits proportional to the log of their variance, and quantized using the optimum Max quantizer. (To be described in the next section). This approach was followed by Woods and Huang 1969 [II-6], which showed results with the Hadamard transform and block sizes of 4, 8, 16, and 256 (full frame) pixels on a side. This paper marked the introduction of subblocks, and the use of block transform coding for images. This is illustrated in Figure 5. The paper by Habibi and Wintz 1969 [II-3] contained a thorough treatment of generalized sampling theory, both for continuous signals and discrete signals[16]. The analysis was based on minimizing the MSE between the original and reconstructed images for a number of different sampling schemes. The statistics were modeled as a Markov process with exponential autocorrelation. The optimum solution was shown to be the Karhunen-Loeve transform for continuous signals and the Hotelling transformation by principal components for discrete signals. This was compared to simple PCM sampling with zero-order hold with very favorable results[17]. Habibi and Wintz also showed the sub-optimum Fourier and Hadamard transforms and plotted the variances of them and the optimum transform as a function of the sample number. All transforms were compared for MSE vs. the number of samples. The work then discussed quantization of the samples (to be discussed in the next section), and further optimized the MSE as a tradeoff between the number of samples and the number of bits used to quantize each sample. The later paper by Habibi and Wintz 1971 [II-4] contained substantially the same

[16] By generalized sampling, they meant transforming the data into a sampled space and representing the data by a finite series of samples. Standard sampling was a simple zero-order hold, i.e., PCM sampling into a set of pixels spanning the image dimension. Orthogonal transform-based sampling was the transformation by a set of basis functions to produce a finite series of samples that can be truncated at any given point to produce a corresponding MSE.

[17] For example, the results for continuous signals show that a series of 12 optimum samples (K-L) produces the same MSE as 40 standard (zero-order hold PCM) samples.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

information, but with further examples of the tradeoff between the number of samples and the number of bits. The 1969 MIT Symposium paper by Pratt and Andrews [II-9] introduced the Fourier and Hadamard transforms for image compression, together with their fast algorithms. The paper discussed quantization both from a statistical point of view as a function of variance and a psychophysical point of view. It proposed the use of more quantization levels for the low-frequency coefficients, and is therefore a precedent of the subjective weighting of coefficients used by JPEG and MPEG. Pratt and Andrews introduced two strategies, zonal coding and threshold coding, for coding the significant coefficients.

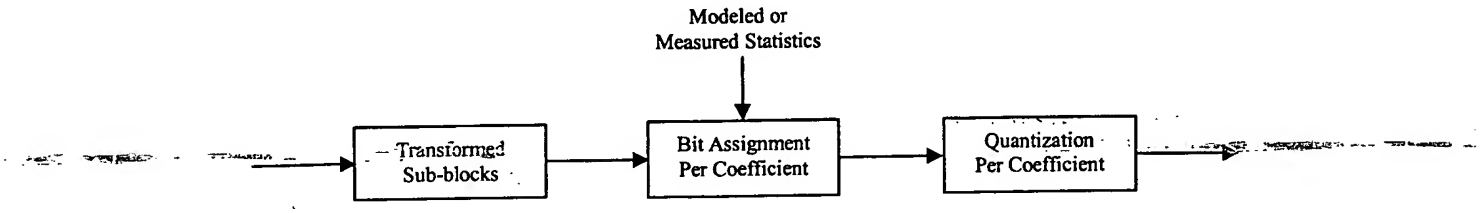


Figure 5. Coding of Transform Coefficients

The zonal coding approach was based on statistical modeling of the variance of the coefficients. In most of the early work, inter-pixel correlation was modeled as a first-order Markov process. Figure 6 shows the row or column correlation matrix for an inter-pixel correlation of ρ_c (typically 0.8-0.95), and a pixel variance of σ_i^2. Using this model, the variances in the transform domain have a non-linear distribution with a maximum variance for the DC term. Coding bits were allocated based on the log of the variances, following the analysis of block quantization by Huang and Schultheiss 1963 [I-19]. The bit allocation matrix was fixed, and therefore known to the decoder without the need to transmit it as overhead. A typical bit allocation matrix is shown in Figure 7.

$$C_i = C_j = \sigma_i^2 \begin{bmatrix} 1 & \rho_c & \rho_c^2 & \rho_c^3 & \cdots & \rho_c^{i-1} \\ \rho_c & 1 & \rho_c & \rho_c^2 & \cdots & \rho_c^{i-1} \\ \rho_c^2 & \rho_c & 1 & \rho_c & \cdots & \rho_c^{i-1} \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \rho_c^{i-1} & \cdot & \cdot & \cdot & & 1 \end{bmatrix}$$

Figure 6. Markov Model Pixel Correlation Matrix.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

8	7	6	4	2	2	1	0
7	6	3	2	1	0	0	0
6	3	3	1	0	0	0	0
4	2	1	0	0	0	0	0
2	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 7. Example Bit Allocation Matrix.

The zonal coding scheme was simple to implement and did not require any transmitted overhead. It corresponded to the psychovisual model. Its weakness was dependency on the accuracy of the statistical model and the stationarity of statistics. It failed badly when there were significant coefficients in the higher frequencies. This happened when there was an event such as a high-contrast edge. To cope with non-stationarity of statistics, the scheme could be made adaptive by modifying the statistics according to local measurement. Tasto and Wintz 1971 [II-10] introduced an adaptive scheme in which sub-blocks of the image were classified into three classes – high detail, low detail in dark areas and low detail in bright areas. The classification was performed using a measure of local activity based on the sum of the squares of the amplitudes of the coefficients excluding the DC term. This parameter was called the "ac-energy". Tasto and Wintz studied optimization of their system in two ways – statistically using the MSE, and by subjective quality. They modeled the error with two components – the truncation error produced by coding only a first subset of the series of coefficients and quantization error produced by assigning a limited number of bits to the retained coefficients. In the statistical optimization, four quantization strategies were compared using three variables – the number of quantization levels (i.e., the number of bits), the level spacing, and the type of codewords used (Natural code vs. Huffman code). The optimum strategy was found to be one that used the same 512-level quantizer for each coefficient; with a Huffman code for each coefficient based the probability of coefficient values falling into each bin of the quantizer. This marked the first time that VLC coding was used for coefficient data. It also for the first time discussed the use of a single uniform quantizer with a scale factor. Both of these form a key part of MPEG coding. In the subjective optimization, Tasto and Wintz studied optimum allocation of the number of levels and level spacing for the three classes of subblocks. They also discovered that a linear increase of quantizer scale factor for higher-frequency coefficients produced the optimum performance. The tradeoff between the number of coefficients truncated vs. the quantization level spacing was however inconclusive. Comparing the MSE minimized and subjectively minimized results confirmed that the human vision sensitivity to distortion decreases both with increased brightness and spatial frequency. Moreover, the insensitivity at higher frequencies is more pronounced at higher brightness.

Reader 1973 [II-11] 1975 [II-12] developed an adaptive scheme using the ac-energy of Tasto and Wintz with three classes of blocks – low energy (i.e., low detail), high energy, and a special class. The special class contained very high energy and corresponded to blocks for which the statistical model failed[18]. Such blocks had a uniform distribution of energy and many significant high-energy coefficients. Visually these blocks contained abrupt discontinuities such as a high-contrast diagonal edge. This approach produced an improvement in quality, particularly because bits could be allocated to the special class to minimize the visibility of artifacts in critical regions of the scene.

Chen 1976 [II-13] extended this approach further into practical systems that were used as products in the first days of video conferencing. In Chen's scheme, four classes of activity were used based on the ac-energy. The classes contained equal populations of blocks, and were allocated bits proportional to total class energy, but from a fixed total number of bits. This scheme therefore produced a constant bit rate on a frame basis. The class of each block was transmitted as side information.

The alternate approach – threshold coding – coded all coefficients whose amplitude exceeded a threshold. Use of a non-zero threshold allowed a measure of noise suppression. This approach had the benefit of being completely adaptive to the data in each block. One problem was the cost of coding the location of the non-zero coefficients. Andrews and Pratt used run-length coding in a simple raster sequence, as illustrated in Figure 8 1970 [II-14]. In the early work this placed the performance of threshold coding on a par with zonal coding.

$$\begin{bmatrix} 34 & -9 & 3 & 0 & 2 & 0 & 0 & 0 \\ 15 & -6 & 0 & 2 & 0 & 0 & 0 & 0 \\ 4 & 1 & 3 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Raster order for run-length coding:

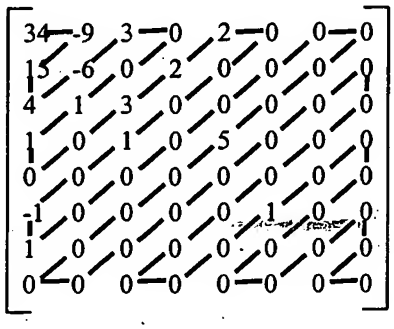
34,-9,3,0,2,0,0,0,15,-6,0,2,0,0,0,0,4,1,3,0,0,0,....

Figure 8. Threshold Coding.

The solution was provided by Tescher 1976 [II-15] in the form of a zigzag scan as shown in Figure 9. This scan traces a path through the zones of maximum variance, approximately ordering the coefficients in decreasing order of variance. The effect is to

[18] In fact, transform coding itself had failed because the expected compaction of energy into a few coefficients had not occurred. Even the Huang and Schultheiss model had failed – the data cannot be decorrelated.

convert the 2D array into a 1D vector concentrating the non-zero coefficients at the beginning of the vector and producing long runs of zeros toward the end of the vector. The zeros can then be efficiently run-length coded. During the development of H.261 alternate scans were investigated, but not found to be useful for that application. For MPEG2 an alternate scan was defined for use with fields of interlaced data. In this case, the doubled spacing of the lines reduces vertical correlation and skews the pattern of maximum variance. H.261 and MPEG2 scanning patterns are discussed later.



Raster order for run-length coding:

34,-9,15,4,-6,3,0,0,1,1,0,0,3,2,2,0,0,0,1,0,-1,1,0,0,0

Figure 9. Zigzag scan for Threshold Coding.

A summary of transform coding was provided by Wintz [II-16] in a special issue of IEEE Proceedings in July 1972. This paper provided a comprehensive overview of transforms and different ways to visualize their effect. In particular, the paper discussed the tradeoffs between companding quantizers and non-uniform quantizers and between non-uniform quantizers with natural codes (FLCs) and uniform quantizers with VLCs. This will be discussed further in the next section. The paper included a brief historical overview, a summary of contributions from various institutions around the world and an extensive bibliography. It is an excellent summary of the state-of-the-art in 1972.

Block Quantization

The decorrelated coefficients must be optimally quantized. The problem was analyzed by Max 1960 [II-17][19] for generalized random variables. Max defined a statistically optimum non-linear quantizer based on the probability distribution of the amplitude of the samples. In summary, the Max quantizer places quantization levels at intervals such that the area under the probability distribution curve is divided into equal intervals, and places reconstruction levels half way between the reconstruction levels on the amplitude scale. This is illustrated in Figure 10 for a Gaussian distribution.

While optimum, the Max quantizer is difficult to use, especially in a real-time adaptive system. The quantization and reconstruction levels cannot be computed simply, so a look-

[19] A similar solution is also reported by Lloyd, but the work is unpublished. Some references refer to the "Lloyd-Max" quantizer.

up table approach is required. However, if it is necessary to vary the degree of quantization as a part of a rate-buffer strategy to achieve a constant bitrate, the levels must be recomputed frequently.

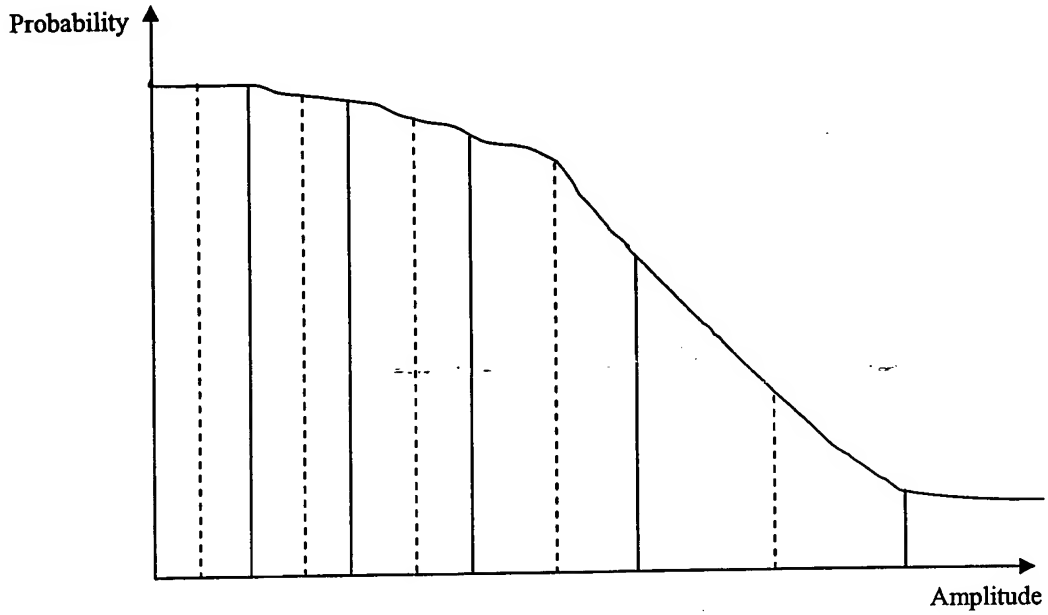
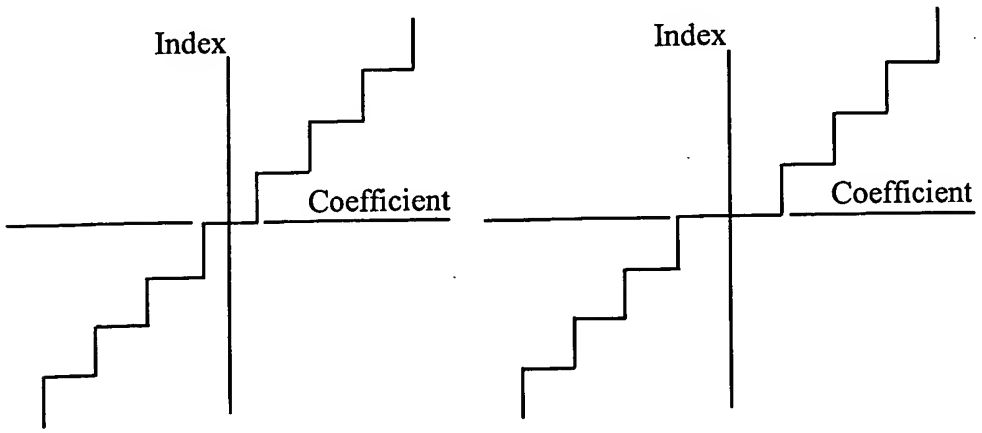


Figure 10. Max Quantizer.

The alternative is a uniform quantizer, preceded by an adaptive scale factor as shown in Figure 11.



a) Intra – Uniform b) Inter – Dead zone

Figure 11. Uniform Quantizer.

This approach was introduced by Wintz and Kurtenbach 1968 [II-18], and subsequently used by Tasto and Wintz 1971 [II-10] and Chen 1981 [II-19] 1984 [II-20]. It is very

simple to implement because the scale factor is merely a truncation to a certain number of bits.
The uniform quantizer was further studied by Brusewitz 1983 [II-21], who analyzed the tradeoff between using the Max quantizer with fixed-length codewords, and the uniform quantizer with variable length codes for several models of the probability density function. This study shows a theoretical improvement between 1dB and 5dB for the uniform quantizer with entropy coding.

Chen also used the uniform quantizer in conjunction with a so-called "dead-zone". This is also illustrated in Figure 11, and comprises a gap around zero that effectively established a threshold on coded values. This serves to reduce the effect of noise and is related to the original threshold-coding scheme.

Perceptually weighted quantization of transform coefficients has been discussed by many researchers. Pratt and Andrews 1969 [II-9] considered more coarsely quantizing high amplitude coefficients and also higher frequency coefficients based on perceptual criteria. Tasto and Wintz 1971 [II-10] provided a detailed analysis of these phenomena. The conclusion is that additional perceptual weighting of quantization values may be applied to any statistically derived weighting without an increase in perceived error. In MPEG the most generic capability is provided by allowing the encoder the freedom to specify any additional weighting.

Entropy Coding

Entropy coding was introduced in 1949 by Fano [I-17] and further developed in 1952 by Huffman [I-18]. Huffman's algorithm has been proven to produce the smallest integer number of coding bits for a given set of samples and associated probability distribution[20]. Entropy codes assign shorter codewords to more probable sample values, thereby achieving a lower data rate overall. This results in a variable-length code (VLC). For transmission or storage, the codes are concatenated into a single bitstream. An important property of these codes allows decoders to synchronize with the bitstream and extract successive codewords. No entropy code is a prefix of another entropy code. Therefore the decoder may unambiguously decode the first sample, and using the length of that sample, it may locate the first bit of the next sample, and so on. Entropy coding does have the disadvantage that the decoding process is relatively complex, and the recursive nature of the code constrains high-speed implementations. Much effort has been expended to find efficient implementations. Another disadvantage is the variability of the data rate and its dependence on the statistics of the data. In a practical system such as MPEG, improbable sample values have codewords that are significantly longer than the original fixed-length code describing the value of the sample. In some cases this leads to data expansion. Notably, when the statistics are not stationary, such as the blocks with a discontinuity, there may be a roughly uniform distribution of energy, and a set of VLC codes optimized for exponential distribution will produce many coded bits.

[20] Arithmetic coding can improve on Huffman coding by the assignment of fractional bits.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

The entropy codes introduced by Fano [I-17], are known as Shannon-Fano coding. These codes may be as efficient as Huffman codes, but are not guaranteed to be so for all conditions. Simple examples are shown in Figure 12, which is extracted from a paper by Connell in 1973 that combines the advantages of Huffman's minimum redundancy code with the fast decoding property of Shannon-Fano codes [II-22]. For MPEG and related standards, these codes are the ones used, and are efficient. The Shannon-Fano code possesses a further important characteristic. The codes of a given bit length have numerical values that are monotonically increasing. This property has been exploited by a number of people to provide efficient decoder implementations.

Probability rank	Huffman Code	Optimum Code	Optimum Code Value
0	10	00	0
1	000	010	2
2	010	011	3
3	011	100	4
4	0010	1010	10
5	1100	1011	11
6	1101	1100	12
7	1111	1101	13
8	00110	11100	28
9	00111	11101	29
10	11100	11110	30
11	11101	11111	31

Figure 12. Huffman and Shannon-Fano Code Comparison. (Connell, 1973)

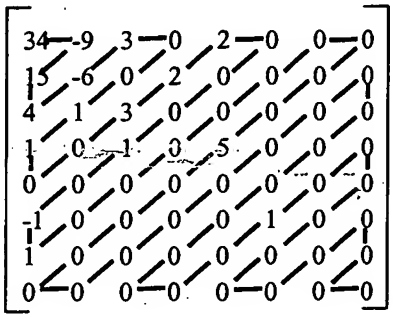
Entropy coding was first considered for image coding by researchers at Purdue University as reported by Tasto and Wintz in 1971 [II-10]. Their work studied tradeoffs between four coding strategies, varying the number of coding bits i.e., quantization levels according to modeled or empirical statistics, varying the quantization level interval, and using either fixed-length ("natural") codes (FLC). They concluded that a combination of a single 512-level quantizer for all coefficients and entropy codes optimized for each coefficient produced the highest performance.

Huffman coding was also used by Chen 1981 [II-19], who defined two VLC tables for coding zigzag scanned coefficients. The first table was used for runs of zeros. The second table was used for the value. Chen introduced a special code – End of Block or EOB – that was used to signal the last run of zeros.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

JPEG uses a Shannon-Fano table strategy in which the VLC encodes the word-length of the coefficient in bits, followed by a natural binary code of the given word-length containing the value of the coefficient. This approach was also used in MPEG to code the value of the DC-term (DC coefficient) in Intra-coded blocks.

A refinement known as 2D VLC coding was introduced in 1986 simultaneously by Chen and Vogel. This is an example of coding using conditional or joint statistics. This is illustrated in Figure 13[21] and is used in H.261, MPEG1 and MPEG2.



2D Run-Level events (not including DC-term):

{0,-9},{0,15},{0,4},{0,-6},{0,3},{2,1},{0,1},{2,3},{0,2},{0,2},{3,1},{1,-1},
{0,1},{9,5},{19,1},EOB

Figure 13. 2D VLC.

Rate Buffering

Most adaptive schemes produce a variable data rate that must be buffered before transmission over a fixed data rate channel or recorded on a digital storage device. Early papers referred to this need without analysis. The problem was first discussed by Bell Laboratories researchers for videophone application. See papers by Mounts 1969 [III-12], Limb 1972 [II-23] and Haskell 1975 [III-15]. The first analysis of this problem for transform coding was provided by Tescher and Cox 1977 [II-24]. The problem was also addressed by Chen 1981[II-19], who credits Tescher with providing the work.

The Scene Adaptive Coder

The various threads of transform coding work were integrated into a system that is widely regarded as the first efficient, practical video coding system using adaptive transform coding. This system, called the Scene Adaptive Coder by Chen was first described in a 1981 ICC paper [II-19], and more famously in a 1984 IEEE paper [II-20]. The scene adaptive coder only performs intraframe coding but otherwise contains all the basic coding tools of MPEG video[22]. In summary these are:

[21] H.263 and MPEG4 use 3D VLC coding in which the last non-zero coefficient is signaled by the third dimension of the joint probability code. This replaces the EOB construct.

[22] And for that matter, JPEG. It is a precursor to Motion JPEG.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

- Adaptive block-based DCT
- Uniform quantization with threshold (similar to dead zone)
- Run-length coding of zigzag-scanned non-zero coefficients
- Entropy coding using a single table for all coefficient amplitudes
- Escape codes for limiting the length of both tables
- EOB (end-of-block) code
- Rate buffer with feedback to the quantizer scale factor

The codec is illustrated in Figure 14.

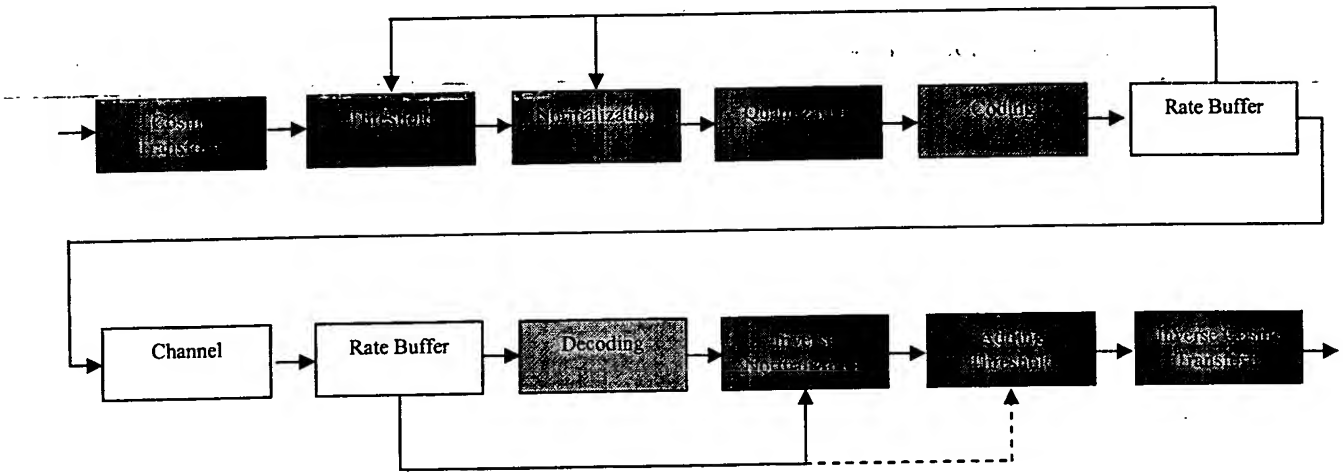


Figure 14. Scene Adaptive Coder. Chen, 1981.

The Scene Adaptive Coder was designed for high performance and efficient implementation. It provided adaptive coding in a one-pass encoding method. The transformed coefficients were first subjected to a threshold that set many coefficients to zero. They were then scaled by a variable scale factor and quantized with a uniform quantizer that could be implemented simply as a rounding operation to a fixed number of bits. This resulted in yet more coefficients being set to zero. The DC-term was coded with a 9-bit fixed-length code. The remaining coefficients were coded in zigzag-scan order using two VLC tables, one to code runs of zero coefficients and the other to code the amplitude of non-zero coefficients. The last non-zero coefficient was followed by an EOB code. This required fewer bits than coding the residual run of zeros. Not all values were contained in the VLC table. Only the first thirty values were VLC coded. The remaining values were coded with fixed-length codes (FLC). These codes were signaled with a special VLC – an escape code. The resultant variable bitrate was smoothed for transmission by a rate buffer. Overflow of this buffer was controlled by feedback to the scale factor. The feedback from the rate buffer was based on buffer fullness and the input rate to the buffer and was adjusted on a block-basis. An empirical curve was used, and

the model attempted to keep the buffer half full at all times. A recursive filter was used to avoid fluctuations. Fill bits were used to avoid underflow.

The paper suggested a number of extensions or improvements. Color could be coded with YIQ components. The I and Q data were subsampled 4:1 in each direction (i.e., 16:1 in area). The coding order of the components was 32Y blocks, 2I blocks, and 2Q blocks. This multiplex was similar to the macroblock construct. Anticipated improvements included deleting isolated coefficients with amplitude 1 or coding isolated zero coefficients as amplitude zero (instead of coding a run). Significant improvement was claimed, especially at low bitrates. The Scene Adaptive Code was built in real-time hardware at Compression Labs. for coding NTSC signals at 1.5Mbits/s

Section II References

1. H Enomoto and K Shibata, "Features of Hadamard Transformed Television Signal", Natl. Conf. of IECE Japan, Paper 881, 1965.

2. H C Andrews, W K Pratt, "Fourier Transform Coding of Images", Hawaii Intl. Conf. on System Sciences, Sep 1967.

3. A Habibi, P A Wintz, "Optimum Linear Transformations for Encoding Two-Dimensional Data", Symposium on Picture bandwidth Compression, MIT, Cambridge Mass., Apr 1969.

4. A Habibi, P A Wintz, "Image Coding by Linear Transformation and Block Quantization", IEEE Trans. on Comm. Technol., Vol. COM-19, No. 1, pp.50-62, Feb 1971.

5. W K Pratt, J Kane, H C Andrews, "Hadamard Transform Image Coding", Proc. IEEE, Vol. 57, No. 1, pp. 58-68, Jan 1969.

6. J W Woods, T S Huang, "Picture Bandwidth Compression by Linear Transformation and Block Quantization", Symposium on Picture bandwidth Compression, MIT, Cambridge Mass., Apr 1969.

7. W-H Chen, "Slant Transform Image Coding", USCEE report 441, May 1973.

8. N Ahmed, T Natarajan, K R Rao, "On Image Processing and a Discrete Cosine Transform", IEEE Trans. on Computers, Vol. C-23, No. 1, pp.90-93, Jan 1974.

9. W K Pratt, H C Andrews, "Application of Fourier-Hadamard Transformation to bandwidth Compression", Symposium on Picture bandwidth Compression, MIT, Cambridge Mass., Apr 1969.

10. M Tasto, P A Wintz, "Image Coding by Adaptive Block Quantization", IEEE Trans. on Comm. Technol., Vol. COM-19, No. 6, Dec 1971.

11. C Reader, "Orthogonal Transform Coding of Still and Moving Pictures", Ph.D. Thesis, Univ. Sussex England, 1973.

12. C Reader, "Intraframe and Interframe Adaptive Transform Coding", SPIE Vol. 66, pp. 108-117, 1975.

13. W-H Chen, C H Smith, "Adaptive Coding of Color Images using Cosine Transform", Proc. of IEEE XII ICC, pp. 47-7 – 47-13, Jun 1976.

14. W.K. Pratt and H.C. Andrews, "Transform Image Coding", USCEE Report 387, March 1970.

15. A G Tescher, R V Cox, "An Adaptive Transform Coding Algorithm", Proc. Of IEEE XII ICC, pp. 47-20 – 47-25, Jun 1976.

16. P A Wintz, "Transform Picture Coding", Proc. IEEE, Vol. 60, No. 7, pp.809-820, Jul 1972.

17. J Max, "Quantization for Minimum distortion", IRE Trans. on Info. Theory, pp.7-12, Mar 1960.

18. P A Wintz, A J Kurtenbach, "Waveform Error Control in PCM Telemetry", IEEE Trans. on Info. Technol., Vol. IT-14, pp. 650-661, Sep 1968.

19. W-H Chen and W.K. Pratt, "Scene Adaptive Coder," Proceedings of International Conference on Communications, pp22.5.1-22.5.6, June 1981.

20. W-H Chen and W.K. Pratt, "Scene Adaptive Coding," IEEE Transactions on Communications, Vol.Com-32, No. 3, pp 225-232, March 1984.

21. H Brusewitz, "Quantization and Entropy Coding", Royal Institute for Technology report, TRITA-TTT-8304, Dec 1983.

22. J B Connell, "A Huffman-Shannon-Fano Code", Proc. IEEE, Vol. [redacted], No. [redacted], pp. 1046-1047, Jul 1973.

23. J O Limb, "Buffering of Data Generated by the Coding of Moving Images", BSTJ, Vol. 51, pp. 239-259, Jan 1972.

24. A G Tescher, R V Cox, "Image Coding: Variable Rate Differential Pulse Code Modulation Through Fixed Rate Channel, SPIE Vol. 119, pp. 147-154, 1977.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

Section III Temporal Coding

Motion-Compensated[23] DPCM

From the earliest days of television, it was realized that the signal was highly redundant in the temporal domain. As noted earlier, Kell in 1929 proposed to transmit only the portions of the scene that changed from frame-to-frame. This concept was explored further beginning in the late 1950s, when it was characterized statistically, and real-time hardware was built to collect the statistics and evaluate coding strategies. Effort was applied to model motion and to measure it in video scenes. Y. Taki, et al investigated the autocorrelation of interframe differences in 1959 along with the distribution of interframe difference signals. In their experiments they picked up a single pel per frame in real-time in an analog way and the amplitude was recorded using a sample-and-hold circuit to give time for their slow measurement equipment. The difference was taken using the sample-and-hold output, and the amplitude was represented in ten levels, i.e. 3.3 bit/sample, in their measurements [III-1].

In parallel, DPCM or predictive coding was being studied as a coding technique in the spatial domain. In particular, researchers at Bell Laboratories were performing extensive studies in pursuit of the Picturephone. These efforts converged in the 1970s when the temporal coding advanced from conditional replenishment to temporal DPCM, and the spatial DPCM was extended to three-dimensional predictors A new development in the 1970s was the combination of transform coding in the spatial dimensions with DPCM in the temporal dimension. This combination was called hybrid coding, and proved to be the successful long-term solution. Initially the hybrid coding was performed in the order of spatially compressing the data with intraframe two-dimensional transform coding, followed by compression in the time domain by predictive coding of the transform coefficients. Meanwhile the pure DPCM approach was developed significantly by the addition of motion-compensated prediction. Most of the efforts were based on the prediction of individual pixels from combinations of pixels in their immediate spatial and temporal neighborhood. The motion compensation was thus limited to very few pixels in distance. Nevertheless, a few efforts described block-based motion-compensated prediction with large displacements capable of describing rapid motion. There were briefly some attempts to apply motion-compensated prediction in the transform domain, but the real breakthrough came, and the two threads of development converged, when the order of transform coding and DPCM in the hybrid coder was reversed. In combination with block-based motion-compensated prediction, the temporal coding was applied to spatial-domain pixels and supports rapid motion with large motion vectors, i.e., displacements. The spatial coding with the transform was then applied to pixel differences constituting the prediction error from the temporal coding.

---

[23] In this paper, the terminology used is "motion-compensated DPCM" or motion-compensated prediction". This is explicit and used in preference to "motion compensation", which is an ill-defined and ambiguous term.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

The timeline is shown in Figure 15 and comprises the following threads of development:

- Temporal statistics, motion measurement, modeling and estimation.
- Pixel-based motion coding.
- Block-based motion coding.
- Motion-compensated transform coding.

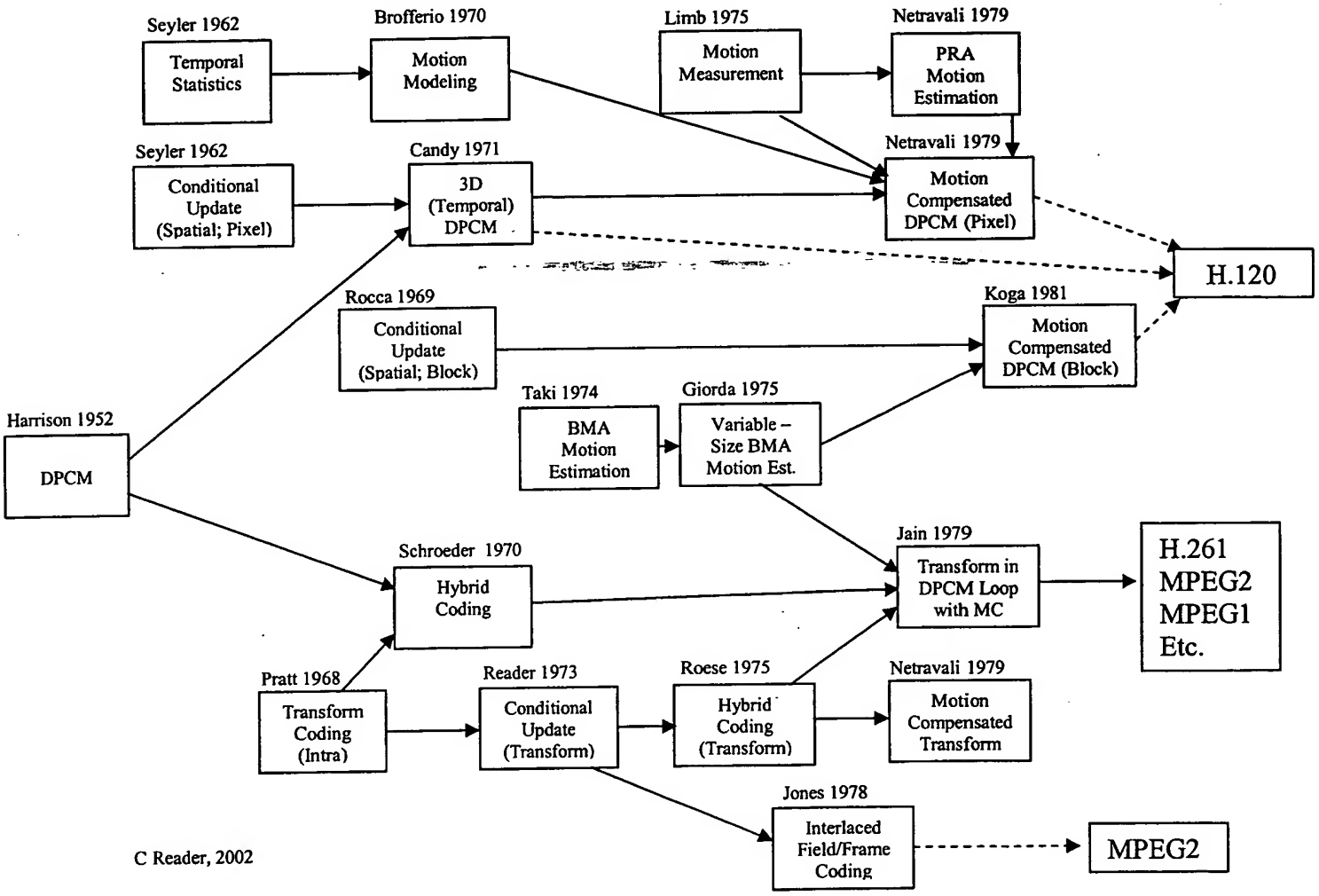


Figure 15 Temporal Coding Timeline (to 1981)

The following techniques became dead-ends. Adaptive 3D (Temporal) DPCM was used in H.120, and was an unsuccessful candidate for H.261. At that point it ceased to be a complete solution for compression. Aspects of it have survived however in the form of AC coefficient prediction in H.263, and H.264/MPEG4 Part 10. Block-based motion compensated DPCM (without transform coding) was last used in the early 1980's videoconferencing systems. PRA-based motion-compensated prediction in both the pixel domain and transform domain were last reported by Biemond in 1987, as discussed below. It was incorporated in a later version of H.120, but no practical use has been

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

reported. Transform coefficient conditional replenishment and DPCM was last reported by CLI (Compression Laboratories Inc.) in 1984. Frame interpolation was featured in a number of the MPEG proposals in 1989. It was not made a part of the MPEG1 or MPEG2 standards. However it has reappeared as direct mode in H.263, MPEG4 Part 2, and H.264/MPEG4 Part 10.

DPCM

Following the invention of DPCM by Cutler 1952 [III-2], it was applied to video coding in 1952 by Oliver [III-3] and Harrison [III-4]. Harrison considered the general case of linear prediction as the weighted sum of previous samples of video and investigated two-dimensional prediction as well as simple prediction along the line. He built hardware to measure the performance of several predictors, using analog delay lines as line buffer memories. Over the next twenty years, researchers at Bell Laboratories explored a wide range of configurations of spatial DPCM. See for example Graham, 1958 [III-5], O'Neal, 1966 [III-6], and Connor, 1971 [III-7]. The latter paper provides a comparison between several techniques and simple previous-element prediction, including an adaptive approach. It therefore provides a good summary of what can be achieved with spatial prediction. This is summarized in Figure 16. From this point on, effort was focused on three-dimensional DPCM, owing to the much higher potential for compression in the time domain.

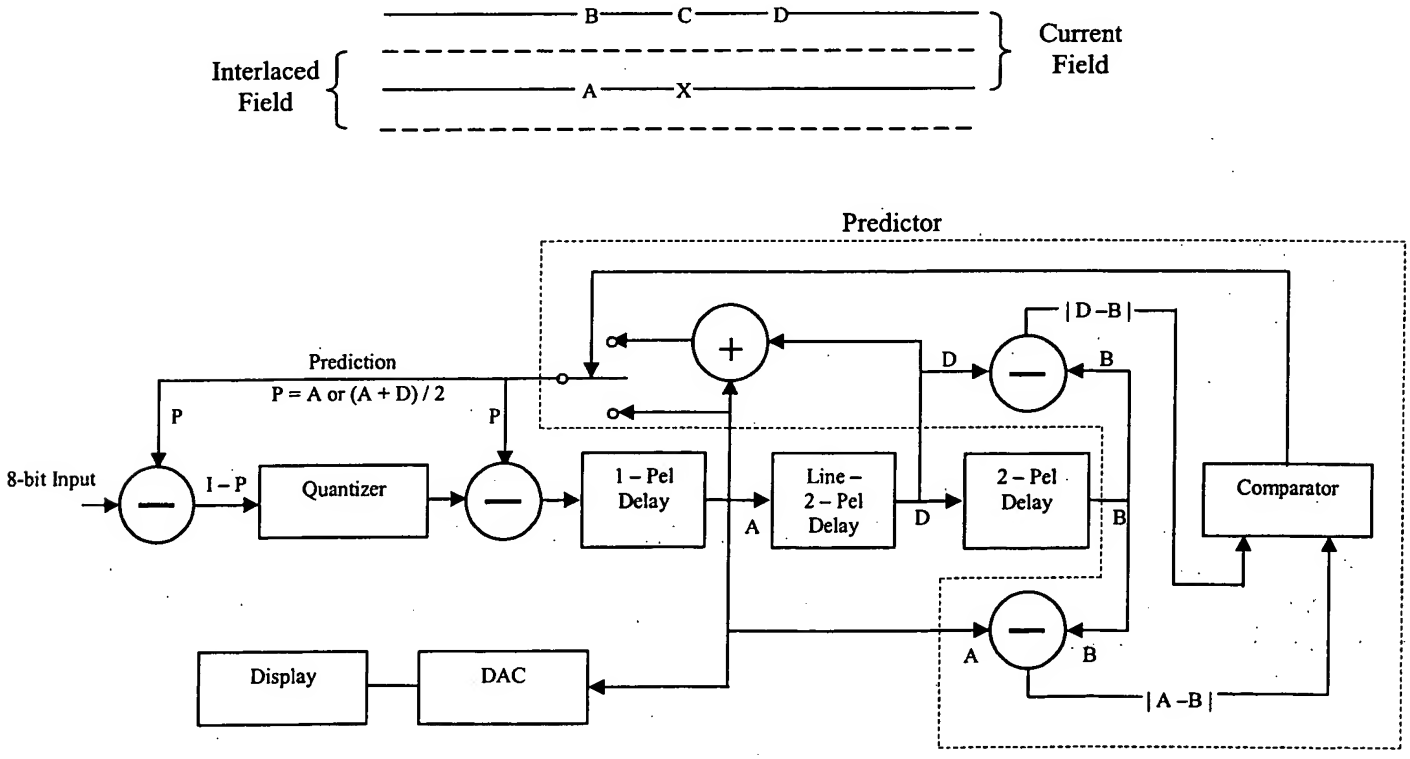


Figure 16. 2D Spatial Prediction, (Connor et al, 1971)

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

Conditional Update, Temporal DPCM in Spatial Domain

The foundation of modern work in interframe coding was laid by Seyler in 1962 [III-8][24]. In this paper, he considered coding interframe differences, recognizing the basic differences between changes due to object motion, camera panning and scene change. He introduced the concept of run-length coding of the difference information and suggested the use of Shannon-Fano coding if appropriate. Key features and issues recognized by Seyler are listed in Figure 17. He suggested that frame difference coding alone was only successful for low motion of objects, noting that averaging the bitrate over periods of high motion, panning and scene changes would require far too much storage. To tackle this, he observed that two effects limit spatial resolution during high motion. First, that the (Vidicon) camera tubes integrated the scene over a frame period, thereby acting as a low-pass filter and blurring detail, Second that the human vision system could not perceive high detail in fast moving objects. At that time, the former effect was the more significant factor. He also observed that the human vision system could not perceive full detail after a scene change. For these reasons, he noted that the resolution of the scene could be reduced during motion, thereby smoothing the data rate. Seyler also described a scheme he called frame run coding, in which frames were skipped at the encoder (with repeated frames at the decoder) until a sufficient amount of change had occurred.

- Coding the Frame Difference Signal (FDS).
- Coding the run-length between differences.
- Shannon-Fano coding of the run-length.
- Rate Buffering.
- Three kinds of frame difference:
  - Moving objects in a stationary environment.
    - FDS works well.
  - Whole scene movement (e.g., panning).
    - Mitigated by TV camera blurring
    - Mitigated to some extent by psychophysical exchange between spatial and motion resolution.
  - Scene change.
    - Mitigated by psychophysical phenomenon of delayed detail perception.
- Low-pass filtering of signal after scene change with variable cut-off frequency.
- Frame run coding – skipped frames based on threshold of movement.

Figure 17. Seyler's Temporal Coder Features. 1962

In a subsequent paper 1963 [III-9] Seyler described a real-time hardware system he built to test the algorithm and measure statistics. He found that for differences up to approximately 50% of the area of the scene, the probability distribution of the differences

[24] Seyler cites three prior references. The aforementioned Kell patent in 1929, a paper by Justy in 1952 and a paper by Schroeder in 1953. In addition, Schroeder was granted a patent in this area in 1940.

is exponential. In a further 1965 paper [III-10], he was able to report extensive testing, and a gamma distribution for the frame differences. Seyler's work provided the foundation for the work of many others and the justification for use of a uniform quantizer for interframe difference signals.

In 1965, Seyler and Budrikis published a significant study of the perception of scene changes [III-11]. This paper confirmed the limited perception of detail after a scene change and characterized it. It was found that resolution could be severely limited immediately after a scene change, with no perception of that limitation provided a minimum of 1/20th of the resolution was provided. Restoration of full resolution could then take up to ¾ seconds without perceived error.

The work on interframe coding at Bell Laboratories was initiated with a paper by Mounts in 1969 [III-12]. This described a simple conditional replenishment scheme on a pixel basis. See Figure 18. The algorithm was implemented in real-time hardware. The addressing of the pixels was line based, with a special code for the first pixel on each line and absolute addresses for the pixels in the line. The coding scheme produced a variable data rate, which was smoothed for transmission using a large buffer equal to two frame sizes. Control of the buffer was effected by varying a threshold that determined whether a pixel had changed or not. Buffer underflow was avoided by reducing the threshold to zero, and buffer overflow was avoided by stopping replenishment when the buffer was full. Aside from the obvious result of picture breakup in the latter condition, the system also demonstrated the "dirty window" effect[25], when motion was high and the threshold was raised significantly to prevent buffer filling.

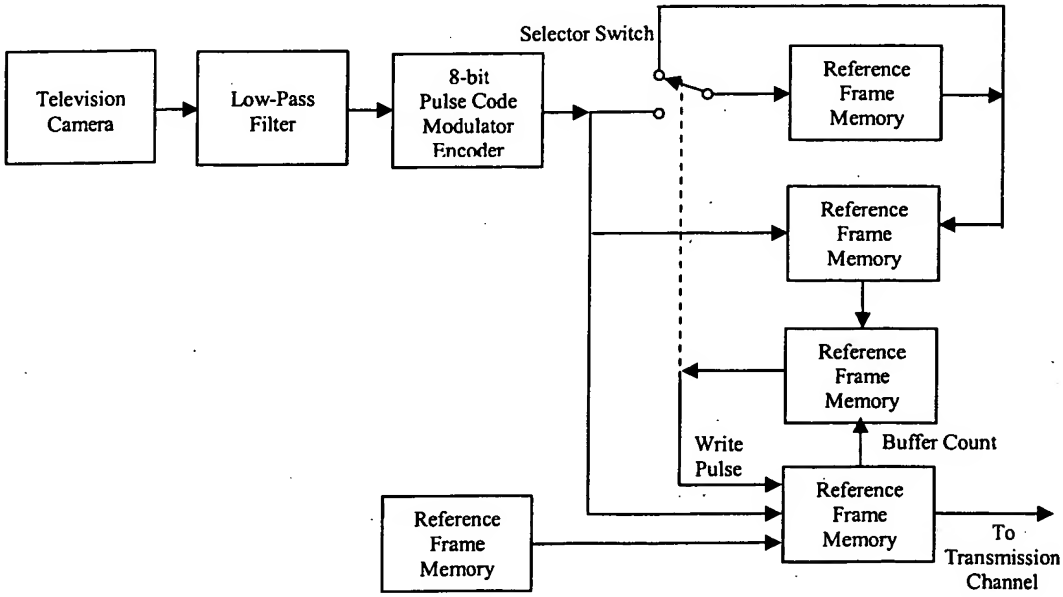


Figure 18. Conditional Replenishment Transmitter Terminal. (Mounts, 1969).

[25] Dirty-window is an effect in which video noise becomes frozen in time and appears like a dirty pane of glass in front of the video scene.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

This work was followed by several papers exploring either the characteristics of this type of system or providing enhancements. See for example, Limb and Pease 1971 [III-13], Candy et al 1971 [III-14], Haskell and Gordon 1975 [III-15], and Haskell 1976 [III-16]. Pease and Limb also published a paper in 1971 [III-17] repeating some of the work of Seyler and Budrikis and reporting on the tradeoff between spatial resolution and temporal resolution in the human vision system.

These papers enhanced Mount's original scheme in several ways. First, the addressing of the changed pixels was improved by using run-length coding. Further, the runs were made more efficient by removing isolated changes, and coalescing runs with small gaps. A unique start code was used for each run, the end of which was also signaled with a special stop code. Second, the pixel difference between frames was transmitted rather than the entire new value. This constituted a shift from simple conditional update to temporal DPCM[26]. An empirical non-uniform quantizer was used with distribution of levels following an approximately exponential curve. The number of bits and levels was limited to cover only the range of the most probable, smallest changes. For larger changes, an escape code mechanism was used with longer code words specifying the absolute value. This is similar to the escape code mechanism used for MPEG coefficients. Third, subsampling was used in active regions to reduce the resolution in correspondence with the camera blurring and psychovisual effects. Subsampling horizontally was reported by Candy et al, and vertical subsampling was reported by Limb and Pease and by Haskell and Gordon. It should be noted that this work was being performed with the Picturephone format, which was interlaced. Due consideration was given to the difference between inter-pixel correlation in the two dimensions for interlaced fields and to subsample the fields to avoid visible artifacts. A different start code was used for subsampled data. Fourth, a forced update scheme was used to ensure that noise would not accumulate in the receiver buffer, and to manage transmission errors. The scheme involved transmitting three lines of data as straight PCM samples each frame. A special start code was used for these lines. The forced update mechanism was also used to manage underflow of the buffer, by transmitting additional lines as PCM. Fifth, the authors anticipated hierarchical coding, in which a simple scheme such as intraframe DPCM was used for local Picturephone service while interframe coding was used for long-distance Picturephone calls. The cost of long-distance calls was assumed to pay for the cost of the frame buffers needed for interframe coding! Sixth, the work considered multiplexing many calls together for long distance, in which bursts of activity on any individual call would be balanced by relative inactivity on many other calls. This anticipated the "stat-mux", i.e., statistical multiplex commonly used today in MPEG for broadcast TV with many channels on a single transmission link or

[26] It should be noted that this change works well for small degrees of motion, such as those in videophone. When motion is large, such as in broadcast TV, simple conditional update will perform better than simple DPCM, because there is little correlation between co-located samples in successive frames. DPCM must be combined with motion compensation in order to perform well.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

transponder. Seventh, Haskell's 1976 paper added temporal subsampling with interpolation at the receiver. This paper also considered motion compensated prediction, which will be discussed in the next section. Figure 19 summarizes these enhancements.

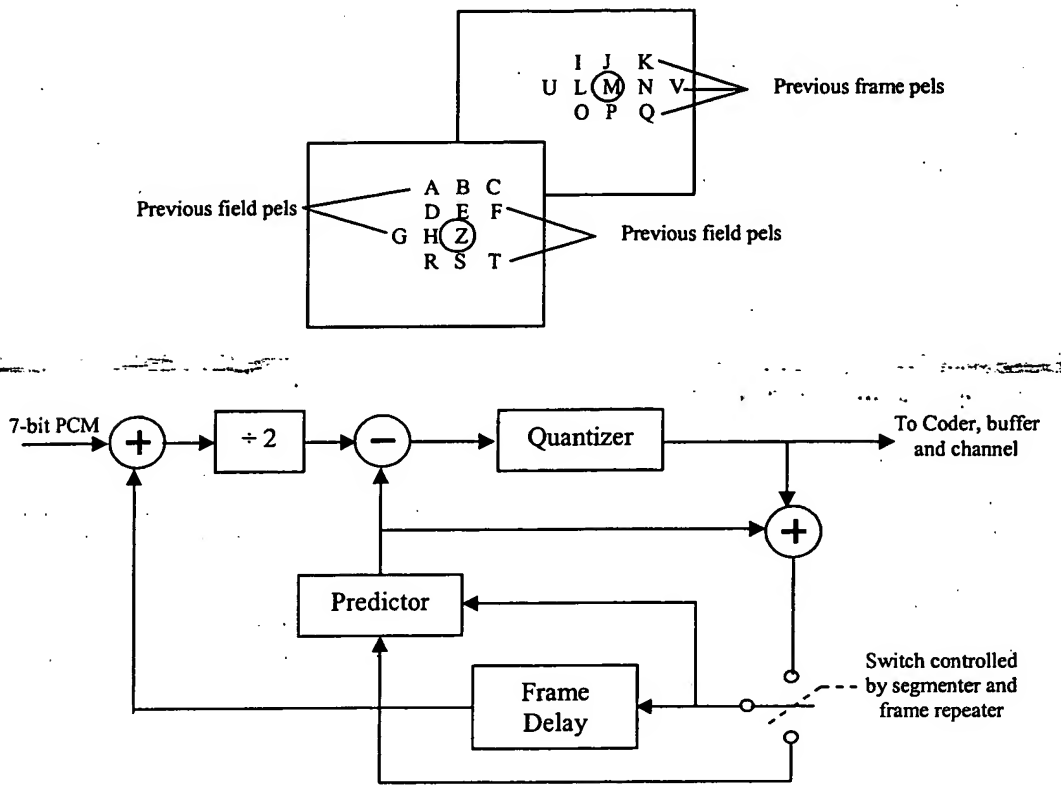


Figure 19. Advanced Adaptive Predictor. (Haskell, 1976).

Beginning in the mid 1970s, researchers in Japan developed systems based on this approach and targeted for videoconferencing. A group at NEC built a series of prototypes and eventually products called NETEC 1975 [III-18] 1976 [III-19]. Similarly, researchers at NTT designed a system called TRIDEC 1977 [III-20]. These systems were flexible and permitted experimentation with different prediction techniques and other facets of the coding scheme. They provided pre- and post-filtering for signal conditioning and noise filtering, spatial and temporal prediction, thresholded movement detection, (sometimes called "conditional update" or conditional replenishment"), and rate-buffer control with spatial and temporal subsampling (interpolation), and forced updating. The systems operated at 6Mbits/s or optionally at 3Mbits/s. Later the performance of the TRIDEC system was enhanced to operate at 1.5Mbits/s 1982 [III-21].

In the late 1970s, a similar effort began in Europe under the name COST 211[27]. This was supported by the European Community, with seven counties participating. The origins and organization of the program are summarized in two papers by Thomson 1981 [III-22] 1982 [III-23]. The focus was on "primary-rate" transmission – 2Mbits/s for Europe. Four countries built real-time hardware and began communicating via satellite in June 1980. One of the designs by GEC/McMichael is described in a 1982 paper 1982 [III-24]. Seven countries participated in software simulations that implemented and optimized the baseline algorithm and experimented with several alternate algorithms 1982 [III-25]. The baseline algorithm was spatial and temporal DPCM, in which selecting whether to use simple temporal prediction or some combination of spatial and temporal prediction was called "switched prediction". The early algorithm is summarized by Chiariglione 1982 [III-26], and contained the same set of features listed above for the Japanese systems. Also described is the complete system with G.732 framing, error correction and encryption. The sync and framing were set up to allow for inter-working with US/Japanese systems based on the NTSC standard. This necessitated standards conversion for line- and frame-rates, and operation at the US primary rate of 1.5Mbits/s. A discussion of spatial and temporal filters, and the beneficial impact of prefiltering data before coding are contained in a 1982 paper by Crawford [III-27].

In the next few years there was essentially a convergence of work between Bell Laboratories in the US, the Japanese, and the Europeans. The algorithm focus remained spatial and temporal prediction, but a push toward lower data rates – 1.5Mbits/s for the Japanese designs, and then 384Kbits/s – required the addition of motion compensated prediction. This will be described in a later section.

A family of standards was created to provide a complete solution for primary rate videoconferencing. A brief overview is provided by Nordquist [III-28]. The COST 211 algorithm was the basis for the H.120 video standard in 1984. H.120 was subsequently revised twice – in 1988 and 1993 – to include (pixel-based) motion compensated prediction, and to include both 625-line and 525-line video. There is no single reference paper describing H.120 Part-3 but there were two papers published which specified it, when combined. Motion-compensated adaptive prediction and VLC of motion vectors and prediction error signals were described in a paper by Koga, 1985 [III-29] (In Japanese). No other paper handled VLC in detail in H.120 Part-3. Background prediction for improving motion-compensated prediction in uncovered background area, making use of background memory updated periodically was reported by Mukawa and Kuroda [III-30]. It is worth noting that motion compensated prediction was combined with background prediction, working in a complimentary manner. There were three prediction modes: intraframe, motion-compensated, and background prediction, one of which was adaptively chosen on a pel-by-pel basis. In effect, adaptive selection was made between intraframe and interframe with another selection between motion-compensated and background prediction. When H.120 Part-3 was implemented in real-time hardware, effectiveness of the background prediction was not as obvious as expected, probably due

[27] Co-Operation in Scientific and Technical Research.

to the premature background extraction algorithm, which is described in the paper. The revisions to H.120 provided enhanced performance, but were overshadowed by H.261, and the push for much lower transmission rates.

Conditional Update, Temporal DPCM in Transform Domain

Schroeder first laid out the concept of hybrid transform-DPCM interframe coding in a patent in 1972[28] [III-31]. Hybrid coding is the combination of transform coding and DPCM. There is no record of the work being simulated or implemented. Curiously Schroeder's main field of work was speech coding. The use of adaptive block transform coding for video was first performed by Reader in 1973 [III-32]. In this scheme, individual frames were first coded by adaptive block coding using the block classification scheme described earlier. Then conditional update was used on the quantized transform domain coefficients to code only significant changes. See Figure 20. When motion compensation is not being used, conditional update works almost as well as DPCM in moving areas, owing to the reduced correlation between co-located blocks. This approach had the advantage that only a fraction of a frame store was required, because each frame was compressed spatially before the temporal coding was applied.

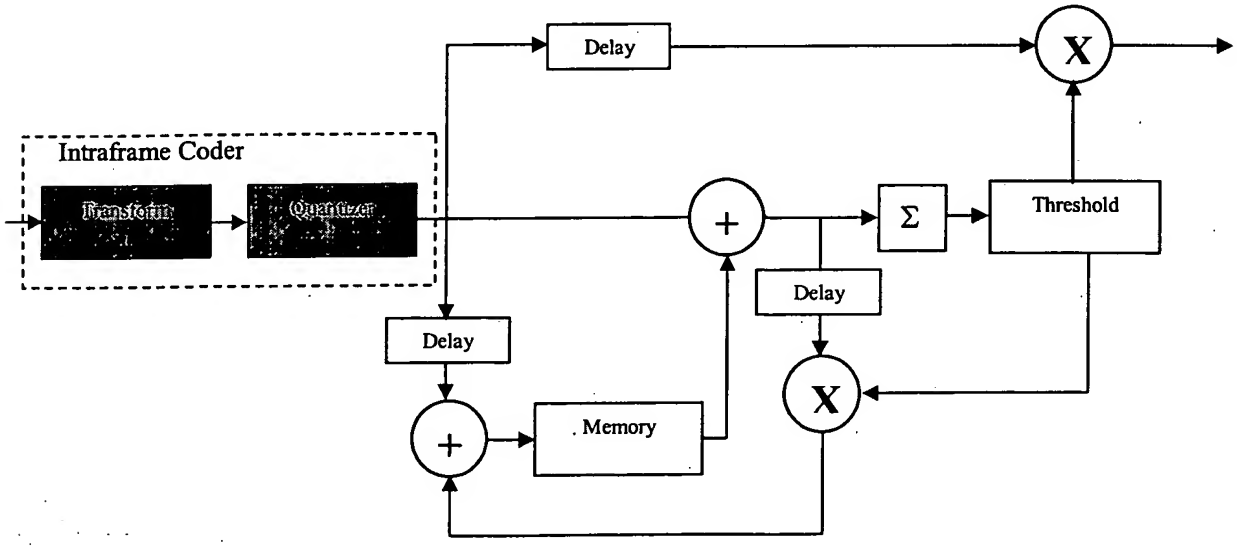


Figure 20. Adaptive Interframe Coder. (Reader, 1973).

In 1974, Habibi introduced hybrid coding in the spatial domain only by transforming one-dimensional blocks of data along the image rows, followed by predictive coding between transform domain coefficients in the columns of the image [III-33]. This is illustrated in Figure 21. The motivation of this approach was cost of implementation, since with a simple first-order predictor it was necessary to buffer only one line of transformed coefficients. This compared favorably with two-dimensional transform coding schemes in

[28] Filed in 1970.

which it was necessary to buffer at least the number of lines in the transform block. The performance was measurably less than two-dimensional transform coding, but the scheme found practical applications such as remote-piloted vehicles (RPV), where size, weight, power consumption and heat issues were at a premium. See for example Whitehouse et al, 1977 [III-34].

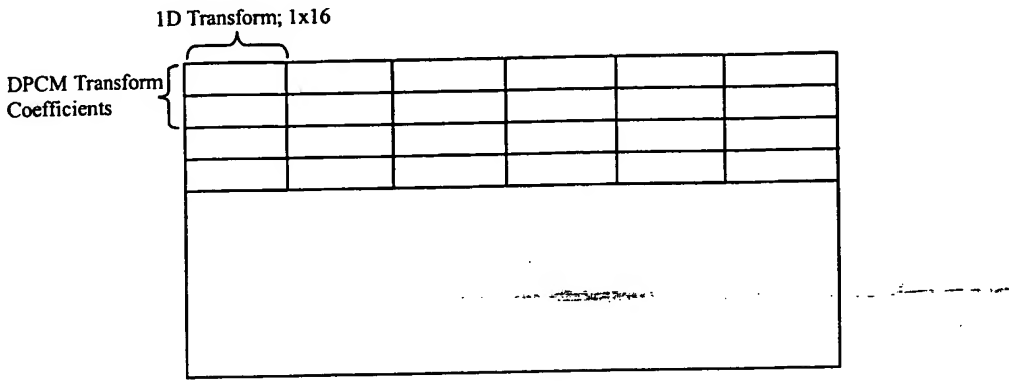


Figure 21. 1D Hybrid Transform-DPCM Coding.

Habibi's scheme was extended to temporal coding in 1975 by Roese et al [III-35]. Two-dimensional block coding was first applied to the video frame, and then predictive coding was used between corresponding coefficients in the succeeding frame. In both Habibi's 1D case and the 2D case, the variance of each transform coefficient was different, so a different quantizer was required for each coefficient. See Figure 22[29]. In accordance with Seyler's finding that the probability distribution of differences is exponential, a uniform quantizer was used. The coefficients were scaled to map onto the quantizer scale. This is a forerunner to the uniform quantizer with scale factor used by MPEG. Roese subsequently extended his work with adaptive block coding, using measured statistics for each block to determine the bit assignment for each coefficient difference 1977 [III-36]. Roese fixed the total number of bits per block to avoid variable data rate problems. As a side note, Roese considered using the Fourier transform, so that motion compensation could be used via the phase-shift property. Roese also compared his work to the use of three-dimensional transforms, and noted they were not practical at that time.

Temporal block transform coding was implemented in a real-time hardware system used for distance learning. Jones described the original system in 1976 [III-37]. The system used the Hadamard transform with 4x4 blocks, and a block classification adaptive coding scheme. In a second system reported in 1977 [III-38], Jones used an 8x8 Hadamard transform with threshold-based change detection, and a rate-buffer feedback control in the same style as the work at Bell Laboratories for pixel-based conditional update. However, when the buffer filled, Jones exploited the frequency property of the transform

[29] Figure 22 applies to both cases. The dimensionality of the transform and the number of predictors are different.

to reduce resolution in the changed areas by transmitting only subsets of the transform coefficients. This was a kind of early form of transform-domain spatial scalability. Jones explicitly selected this approach over the alternative approaches of reducing the number of bits per coefficient (i.e., SNR scalability), or reducing the number of updated blocks.

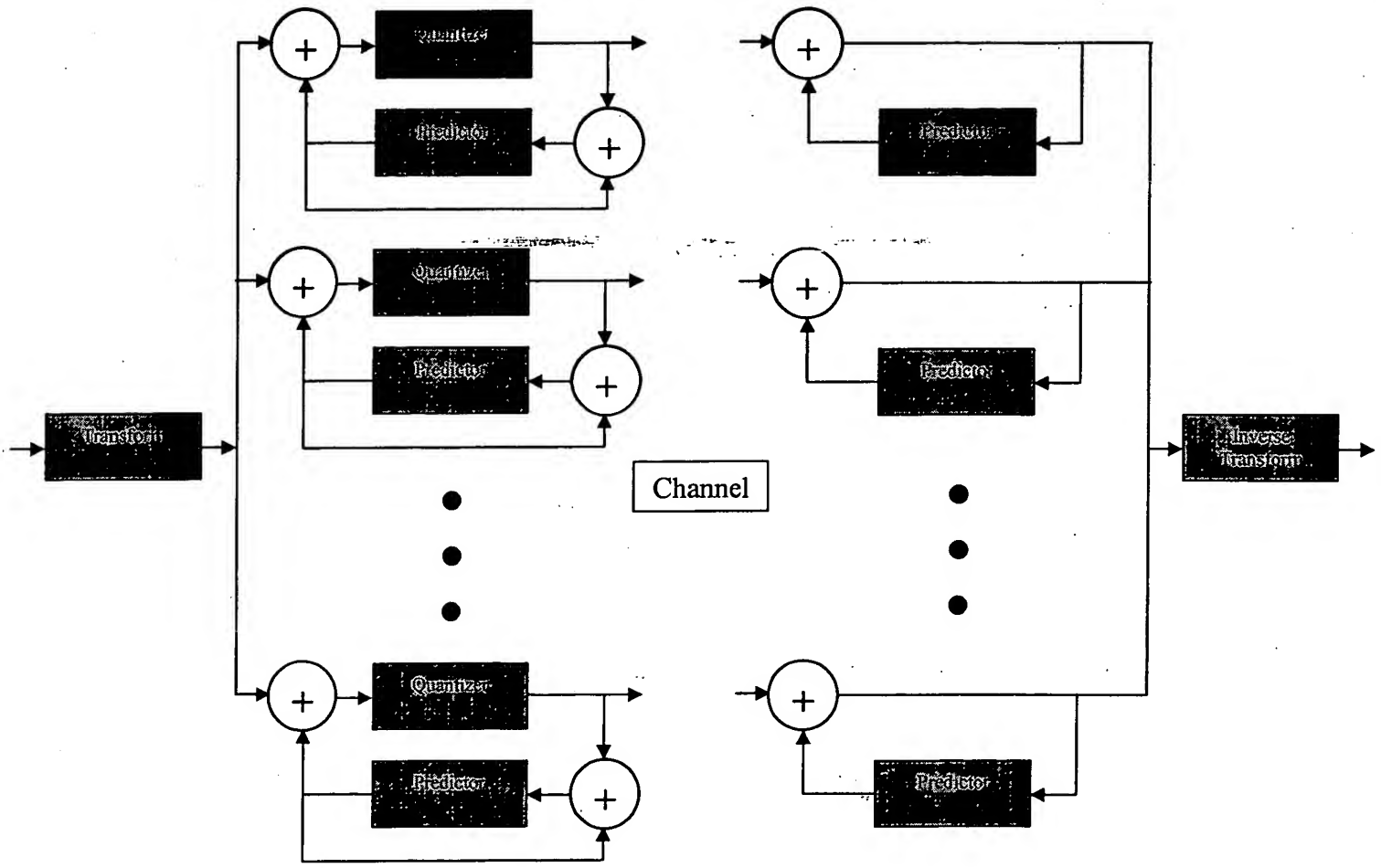


Figure 22. Hybrid Transform/DPCM Coder. (Habibi – 1D; Roese – 2D).

The hardware systems built by Jones and colleagues operated on standard NTSC video signals. They processed data on a field basis to minimize the amount of buffering required. Nevertheless, Jones noted the problem of reduced vertical correlation in fields and in 1978, published a paper that examined the performance of field coding versus frame coding in which the two fields were interleaved and processed as a single frame [III-39]. For the material tested, frame compression produced a lower mean square error, but produced blurring on edges perceptually, as would be expected. This is the first

example of field vs. frame DCT coding, and was considered only on a whole field/frame basis, not on a block-by-block basis.

Motion-Compensated Prediction – Motion Estimation

Several threads can be identified in the development of motion-compensated prediction. (Refer to Figure 15). First, motion was modeled and techniques were developed to measure motion. This led to the general area of motion estimation. Second, pixel-based methods were invented as an extension of conditional update and temporal DPCM. Third, block-based methods were introduced, essentially for simplicity of implementation. In general the pixel-based methods did not require the transmission of motion vector information – the decoder would perform the same set of operations as the encoder to produce the prediction. The pixel-based methods used luminance gradients to determine the direction and distance of motion. In the early work it was necessary to segment the scene into stationary and moving parts, and if there were multiple objects moving independently, to further segment the scene into those objects. The block-based methods required side information to be transmitted to describe the motion vector. In some of the work, a single motion vector was selected for all the blocks found to be long to the same moving object. The first paper on motion-compensated coding was by Rocca in 1969 at the MIT Symposium [III-40]. This paper described an arbitrary-shaped block-based motion-compensated technique, with one motion vector being transmitted for each region of points (i.e., block). The paper rationalized a maximum motion vector size of 10 pixels, based on psychovisual considerations and rationalized a maximum block size of 20x20 pixels as a tradeoff between the number of bits for coding the motion vector and the number of bits for coding the prediction error. These arguments correspond well with modern experience.

Motion Modeling and Motion Estimation

Rocca and colleagues published a series of papers in the 1970s measuring and modeling motion. See Rocca and Zanoletti 1972 [III-41], Caffario and Rocca 1976 [III-42], Brofferio and Rocca 1977 [III-43] and Caffario and Rocca 1979 [III-44]. Initially, they developed a model based on the assumption that the image could be divided into a set of contiguous polygonal zones of uniform brightness, each of which possessed its own motion. This was then modeled as a Poisson process in order to develop statistical characteristics that matched those of the image data, i.e., uniform amplitude distribution and exponential spatial and temporal covariance. It was claimed that matching the contour lines (of equal brightness) was an easier method of motion estimation than correlation by block matching. The optimum block size was shown to be 70-80 pixels. This is close to the 16x16 prediction block size used in basic MPEG coding. Coding was performed by transmitting displacement vectors and quantized differences. Subsequent papers used linear regression techniques with spatial gradients to estimate motion. The techniques could only track motion for 2-3 pixels, but the accuracy was as high as 0.1 pixels. For coding the data, the papers explored a number of options, including the Max quantizer, a log quantizer and a conditional entropy quantizer in conjunction with Shannon-Fano coding. In later papers, they segmented the image into three areas and

used different coding techniques for each region – conditional update for background with little or no motion, motion-compensated prediction for moving objects, and intraframe prediction for areas best coded using conventional spatial DPCM. The latter case would apply to areas of uncovered background for example. Uniform quantization was used in the temporal and motion-compensated modes while non-uniform quantization was used in the spatial coding mode. Run-length coding was used for runs of zeros for prediction error data, together with codes for the prediction errors and two special codes to change mode. The schemes also considered sub-pixel accuracy for the motion with bi-cubic interpolation, and the benefit of using low-pass filtering both to improve the segmentation into moving and non-moving areas and the motion estimation. While the approach of segmenting the scene into stationary and multiple moving areas, and using a single motion vector for each area could be successful for a small number of rigid objects moving in the imaging plane, it was noted that for complex motion, a block-based approach with a motion vector for each block should be used.

In the same period, Limb and Murphy presented two papers measuring motion in images 1975 [III-45] 1975 [III-46]. The technique used was to accumulate inter-pixel differences in both space and time, then apply certain operators to them to estimate the speed and direction of motion. If there were multiple moving objects, or an object with complex motion, it was necessary to segment the image into areas with uniform motion. These techniques were shown to work well for small degrees of motion – up to 2-1/2 pixels per frame.

These approaches were basically motivated by a legacy of processing each pixel successively in the spatial domain, and sought to extend the fundamental concept of pixel-based DPCM into the time domain. The work was also focused on the Picturephone application, which was characterized by a simple content model – the head and shoulders view against a static background – and small degrees of simple translational movement. Netravali and Robbins 1979 [III-47] provided a comprehensive review[30] of this area and developed a sophisticated yet simple to implement motion estimation technique. The work is an extension of the approach by Limb and Murphy 1975 [III-46], and Cafforio and Rocca 1976 [III-42]. This was called the Pel-Recursive Algorithm (PRA), and used a simple local area operator to iteratively compute the motion vector from one pixel to the next[31]. This provided a way to adapt to local variations in movement, including multiple objects and complex motion. The scene was segmented into three regions – no motion, motion requiring update information ("uncompensable regions"), and motion for which update information was not necessary ("compensable regions"). The two types of region with motion correspond to the MPEG modes of "Coded" and Not-Coded". A coded region is one in which prediction error data is required in addition to the motion vector, while the not-coded region is accurately represented with the motion vector alone. Netravali and Robbins also studied the issue of sub-pixel accuracy, and considered

[30] The paper discusses four techniques for motion estimation – i) point-by-point, ii) pattern matching, iii) spatial and temporal differences iv) adaptive linear prediction.
[31] The algorithm is a steepest-descent method.

accuracy up to 1/128th pixels. They found there was no improvement beyond 1/8th pixels. The work was also performed with interlaced data, taking into consideration the different correlation between lines of a field. An important aspect of this algorithm was that the motion information was not transmitted, but rather computed at the receiver in the same way as at the encoder. There was no side information therefore for motion vectors. However, it was necessary to transmit the segmentation information for the three regions.

The techniques of Rocca et al modeled the scene as a set of constant-brightness zones having polygonal shape and arbitrary size. These zones would move from one frame to the next, and in general would experience a lifespan of birth, growth in size, decay and extinction. The zones were tracked from frame-to-frame by motion vectors and difference values were transmitted to correct for errors. The techniques of Limb and Murphy and Netravali and Robbins, were an extension of simple pixel-based linear prediction, to include an offset in the temporal prediction to account for motion. These techniques sought to avoid the massive computational burden of cross-correlation techniques, yet were either complex to implement or were very limited in the range of motion they could exploit. A much simpler approach, using block matching of regular, square blocks was introduced in 1974 by Taki et al [III-48], and Brofferio et al [III-49] See also Giorda and Racciu 1975 [III-50] . This approach, which forms the basis of MPEG coding, divides the frame into contiguous, square blocks and finds a best match for each block in the current frame, by a (generally) offset block in the previous frame. Figure 23 illustrates the technique, which was called the block matching algorithm (BMA). The example shows 4x4 block matching with +/- 2-pixel motion. The upper-left pixel of the matched block can lie anywhere in region D. The Giorda paper describes the use of the sum of absolute differences (SAD) metric to find the block with maximum correlation, and the use of a threshold on the minimum SAD to establish that there is adequate correlation between the two blocks. This is the same as the Intra/Non-Intra decision used in MPEG. The threshold was made a function of both brightness and degree of motion, acknowledging the psychovisual limitations to perception of error at low brightness levels and high motion. Giorda and Racciu analyzed the performance of the system with different block sizes for a range of motion. They found that a block size of 4x4 was optimal for violent motion, while a size of 8x8 was best for smaller amounts of motion. Therefore they adopted an adaptive scheme in which 8x8 motion estimation was tried first. If it failed for a block, then that block was subdivided into 4x4 blocks. If it failed for any of those blocks, a simple intra mode of coding was used. Unsurprisingly, they found such events typically occurred on the edges of moving objects. This is the first example of adaptive block size for motion compensated prediction.

Over the next ten years, a number of efforts explored ways to efficiently compute block-based motion estimation. Giorda and Racciu described an exhaustive search procedure, but they only considered displacements of +/- 4 pixels. This might be acceptable for head-and-shoulders videophone applications with reduced resolution image size, but not for broadcast TV use, especially for sports. Performing exhaustive search motion estimation on NTSC/PAL resolution frames with motion vectors as large as +/- 64 pixels requires billions of operations per second. Therefore various sparse search techniques

were explored. Jain and Jain 1979[32] [III_51/1981 [III-52] developed a logarithmic search procedure. Koga et al 1981 [III-53] described a 3-step search procedure with subsampled data for further reduction in computation, and Ninomiya and Ohtsuka 1982 [III-54] described an iterative procedure starting with the displaced location from the previous field. Kappagantula and Rao 1983 [III-55] described a search procedure that combined features of the Jain and Koga algorithms, and Srinivasin and Rao 1984 [III-56] produced a "one-at-a-time" algorithm that optimizes the search for each direction. Bierling developed a hierarchical technique based on the Koga 3-step approach but using low-pass filtering at each step [III-57]. These techniques are illustrated in Figure 24.

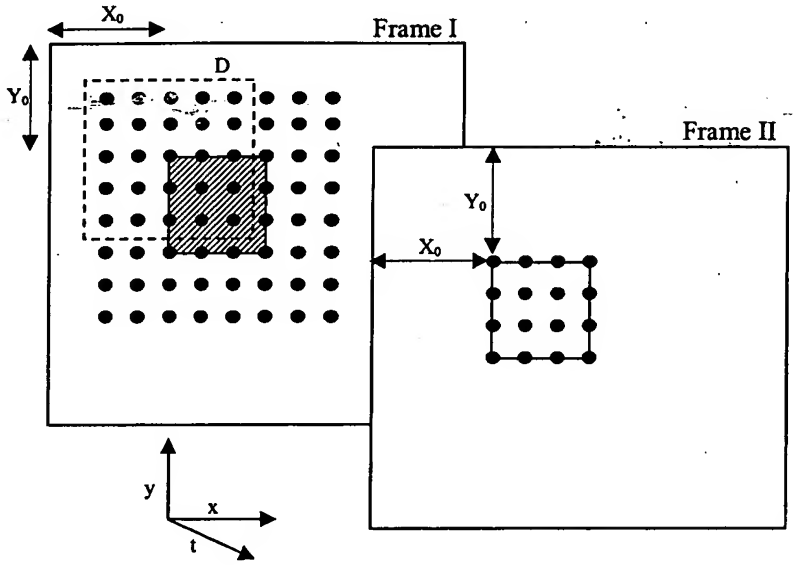
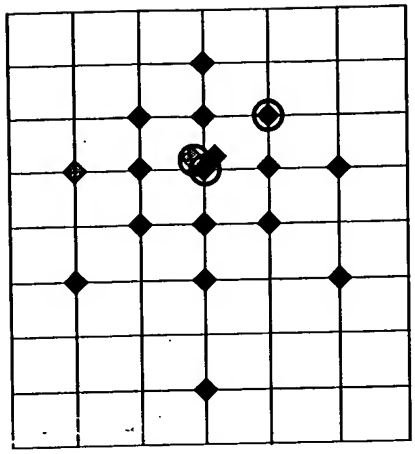


Figure 23. Example of Block Matching. (Giorda & Racciu).

Some other approaches were also tried. Stuller and Netravali 1979 [III-58] extended the PRA technique into the transform domain. Bergmann 1982 [III-59] used a simplified cross-correlation technique based on the zero crossing of the derivative to find the peak. Hsing 1987 [III-60] developed a technique combining aspects of PRA and BMA. Thomas 1987 [III-61] applied classical Fourier correlation techniques to the BMA approach. All of these efforts were strongly focused on minimizing the cost of implementation for use in business or consumer products. It should be noted that there were other applications having far less stringent constraints on cost or physical size, and that sophisticated image and video correlators have been developed for professional TV standards converters and military intelligence applications.

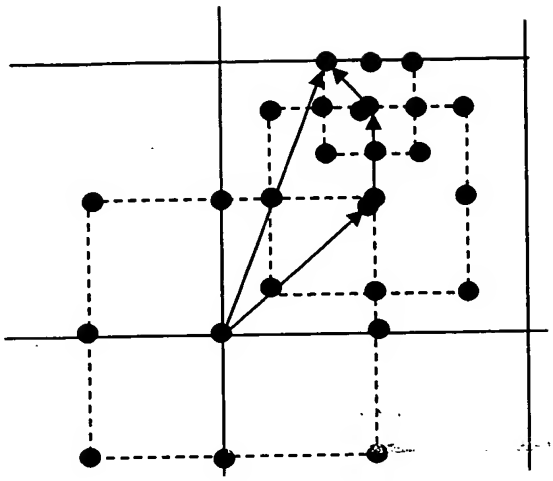
[32] First presented at PCS 1979.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

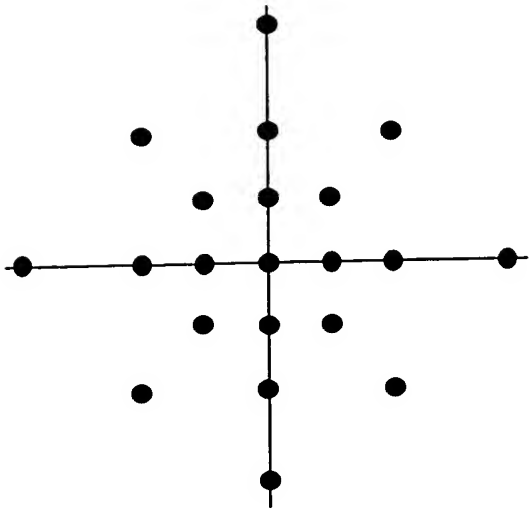


◆ First iteration
◆ Second iteration
◆ Third iteration
○ Winner

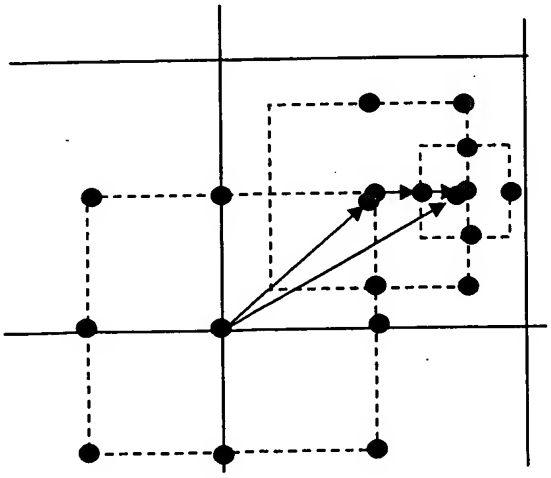
a) Jain & Jain, 1981



b) Koga et al, 1981

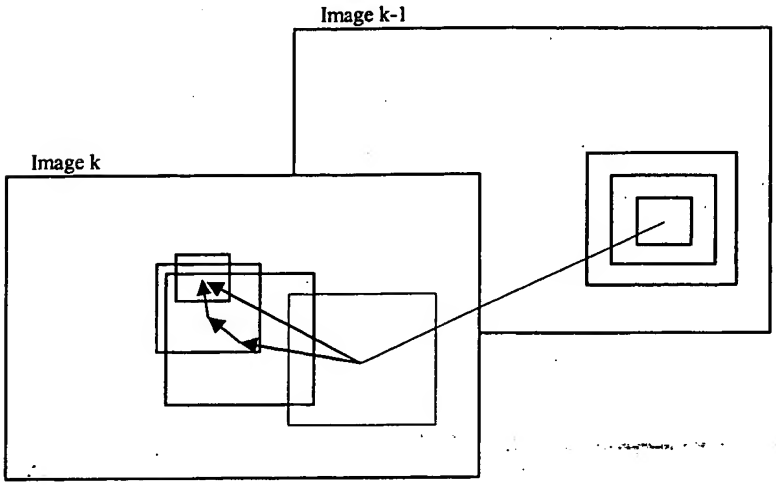


c) Ninomiya & Ohtsuka; 1982
(As used by Ericsson; 1985)



d) Kappagantula & Rao; 1987

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com



e) Bierling; 1986

Figure 24. Fast Motion Estimation Algorithms.

Pixel-Based Motion-Compensated Interframe Coding

Two threads of development continued based on PRA and BMA motion estimation. These developments were now related to real product development and the videoconferencing market began to emerge. Performance of the codecs and sophistication of the algorithms advanced in steps from approximately 3Mbits/s to 1.5Mbits/s (T1 rate), then to 384Kbits/s and finally to 64 or 56Kbits/s. This was driven by the demands of videoconferencing over narrower standard digital communication links.

Further development of the PRA techniques was limited, probably due to the complexity of implementation. As originally developed, PRA was applied to scenes with a single moving object against a stationary background. The technique became much more complex when scenes with multiple moving object and camera motion are considered. The work of Cafforio and Rocca 1976 [III-42] and Netravali and Robbins 1979 [III-47] on PRA techniques was extended by Walker and Rao 1984 [III-62] and Biemond et al 1985 [III-63]. The latter applied Wiener linear estimation techniques to improve the speed and accuracy of the recursion. Even so, the authors noted that PRA techniques could be used only for small motion of up to two pixels per frame. While this is useful for videophone applications, it is too limited for broadcast TV. Considerably more effort has been expended on the BMA approach.

The NETEC products developed by Koga et al at the NEC Corporation initially used simple interframe coding and operated at rates of 6Mbits/s down to 3Mbits/s [III-64]. In order to further lower the bitrate to 1.5Mbits/s and hence enable transmission via T-1

links, it was necessary to add motion compensation. This is described in a 1981 paper by Koga et al [III-53], and is illustrated in Figure 25. Motion estimation was performed with the 3-step algorithm noted earlier. The paper noted that motion vector data was non-uniformly distributed, and therefore should be entropy-coded. This is similar to MPEG, but the paper did not report differential motion vector coding. Analysis of optimum block size determined there was not a strong relationship to coding efficiency, and a block size of 4x8 or 8x16 was suggested. Real-time hardware was built and tested. It was found the efficiency of motion-compensated coding was about two-times better than simple conditional update coding at low bitrates of 1.5Mbits/s, and three-times better at higher bitrates such as 3-6Mbits/s.

In a later, 1983 paper [III-65], Koga et al reported on improvements. They mentioned that the prediction errors were entropy coded as well as the motion vectors. The motion vector was differentially coded with respect to the previous block[33]. This is the same as MPEG. They also studied the statistics of motion vector differences and prediction error differences and suggested run-length coding of zero-difference motion vectors and prediction errors, with Huffman coding of non-zero motion vector and prediction error differences. The scheme also jointly optimized the coding of motion vector and prediction error difference data, resulting in modest improvements in coding efficiency. This system was also built in real-time hardware, and used for subjective testing (NETEC-X1). Reported results indicated significant improvement over non-motion-compensated systems.

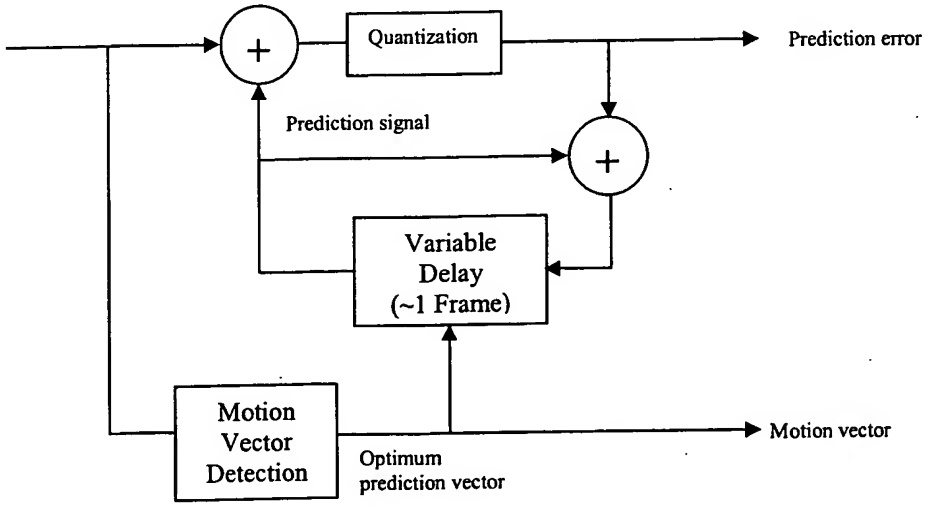


Figure 25. Motion-compensated prediction encoder. (Koga, et al, 1981).

[33] The authors also considered previous field or frame motion vector difference, but found little distinction.

A subsequent paper in 1985 [III-66] reported a further improvement in which the coder switched adaptively on a pixel basis between motion-compensated interframe prediction and intraframe prediction. This was called the NETEX XV system, which supported bitrates of 512Kbits/s, 1Mbit/s and 2Mbits/s. The system operated at broadcast TV resolution and coded the color information into the luminance blanking interval. A summary of all the work can be found in another 1985 paper [III-67].

At the same time, researchers at NHK also investigated the BMA approach. Ninomiya and Ohtsuka described their work in two papers in 1982 [III-54] and 1984 [III-68]; the latter paper comprised an extension of the technique to color. As noted before, an iterative method was used for motion estimation, with the corresponding motion vector from the previous frame providing the initial estimate. Sub-pixel accuracy was also used. An experimental real-time system was built to evaluate this scheme. Like MPEG, the coding scheme was multi-modal, but the modes were rather different. There were two prediction modes – normal motion-compensated prediction (used most of the time), a special second-order temporal prediction used only infrequently during fades or cross-fades, and an intra-field DPCM mode to be used when prediction failed. There were four modes of sampling – normal sampling, mixed sampling, subsampling and field-repeat.

The normal prediction mode used a block size of 8x16 pixels. Note that the scheme operated on interlaced data, and this was the same as the 8x16 mode of MPEG2 prediction. The authors found a problem with the "dirty window" effect for blocks that overlapped both a moving object and a stationary background. They chose to implement a per-pixel decision whether to use motion-compensated prediction[34]. The fade-prediction mode was a simple field-averaging approach, and was selected if it produced lower prediction error than the normal mode, with some hysteresis being used in the switching decision. The sampling modes were switched on a field basis, depending on the degree of motion, except for the mixed mode, which was switched on a block basis. This followed the well-established procedure that it is better to subsample than raise the threshold very high owing to psychovisual and camera-blurring effects[III-8][III-15]. The decoder used signals from the previous frame for subsampled points in stationary areas and interpolated points from the current field in moving areas. A multi-modal quantizer was used, in conjunction with thresholding and companding designed to minimize the dirty-window effect. Reference was also made to field prediction vs. frame prediction without complete explanation. The selection of coding modes and threshold levels was based on the rate buffer status and current field volume of coded bits. Formal subjective testing was performed at 1.6 bits/pixel or about 14.5Mbits/s.

Rivero 1985 [III-69] evaluated the various motion estimation schemes and chose the 3-step algorithm by Koga et al for further development. The search pattern was modified for use with interlaced data, by considering the different line-to-line correlation within a field. The prediction operated with three modes – conditional replenishment, temporal

[34] This is a useful feature that can be implemented in a purely spatial-domain algorithm, but could not be used in a transform-domain approach such as MPEG.

prediction and spatial prediction. However, the meaning of conditional replenishment was that differences below a threshold were not coded. So in effect, this was the same as temporal prediction with a threshold, and is used in MPEG for interframe data, where the quantizer has a dead-zone. Spatial prediction was used when temporal prediction failed to produce a good result.

Transform-Based Motion-Compensated Interframe Coding

In 1979, Netravali and Stuller [III-70][III-58] attempted to extend the PRA technique to the transform conditional update and DPCM schemes of Reader and Roese. Their algorithm predicted each transform coefficient from the same coefficient in the transform of either the collocated block or motion-compensated block in the previous frame. The decision was based on which choice was better for the previous coefficient in the block. The authors also tried this approach with the Limb and Murphy BMA motion estimation. The PRA motion estimation did not require transmission of side information, while the latter required a transmitted motion vector. They tried several transforms (Hadamard, Cosine, etc.) without much difference. They also tried several small block sizes.

Also at PCS in 1979, Jain and Jain took a critical step when they inverted the order of the DPCM and transform coding in the hybrid coder to produce the familiar arrangement used in MPEG. The work was published as an internal report in 1979 [III-51] and further in 1981 [III-52] together with the logarithmic motion estimation scheme discussed earlier and a motion-compensated hybrid coder with the transform outside the DPCM loop. This is shown in Figure 26, from which it can be seen that when motion compensated prediction is added to the original hybrid coder, three transform blocks are necessary in order to maintain the state of the prediction frame buffer. The authors stated that they moved the transform inside the DPCM loop as a simplification. This has become the standard configuration for motion-compensated hybrid coders. The authors also described frame interpolation, as discussed in the next section, and the paper was therefore a landmark in the development of MPEG compression. The coding scheme used however, was still based on adaptive zonal coding. It used a variance-based bit allocation, Max quantizer and classification of blocks into classes. The scheme produced a variable bitrate, necessitating a rate buffer. No details were given for this. The results shown indicated that interframe coding without motion compensation performed about 2.5dB better than intraframe coding at 2bits/pixel. Motion compensated prediction added a further 7dB.

Motion-compensated interframe hybrid coding was compared with motion-compensated interframe DPCM in a 1985 paper by Furukawa, Koga and Niwa [III-71]. The transform coefficients were quantized with a uniform quantizer having both a variable threshold and variable scale factor. This quantizer was thus similar to the one used by MPEG. The paper concluded that the hybrid coding produced a sharper, higher resolution image than DPCM coding for the moving image and handled scene change better. The DPCM coding performed better on uncovered background areas. Rates tested were 0.1-0.3 bits/pixel[35].

[35] The section on performance relates rates in bits/pixel to picture formats and rates in bits/s.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

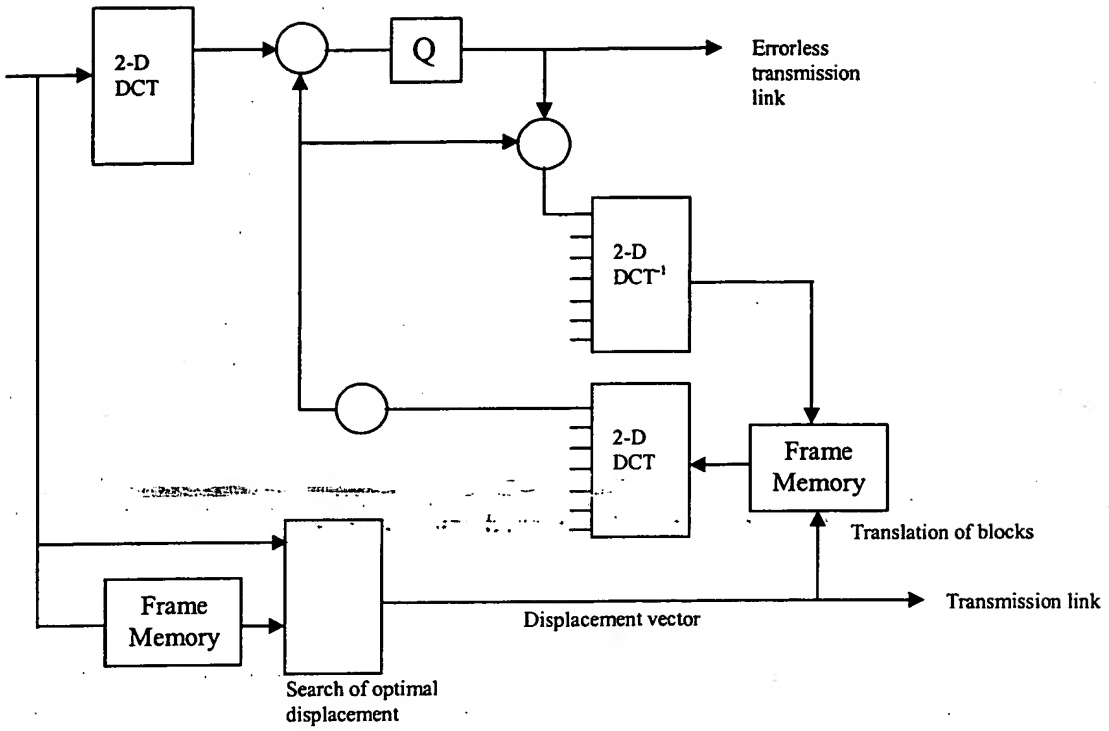


Figure 26a. Interframe hybrid coding with motion compensation. (Jain & Jain, 1981).

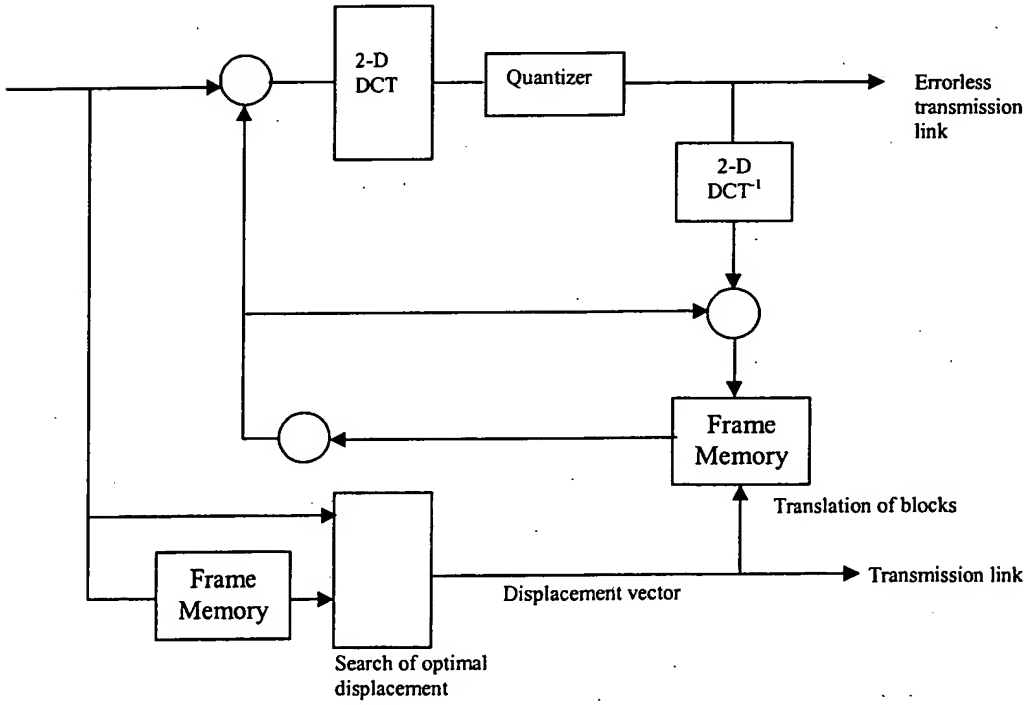


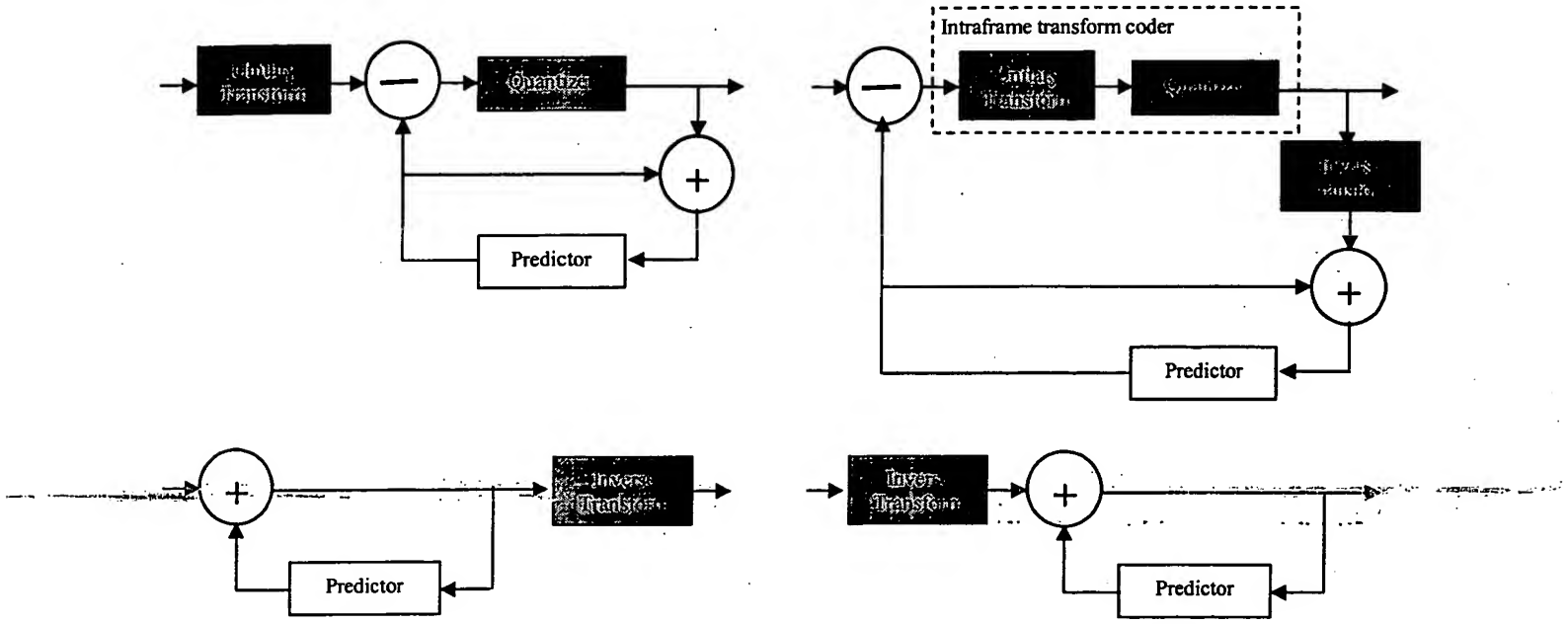
Figure 26b. Simplified Interframe hybrid coding with motion compensation. (Jain & Jain, 1981).

Ericsson 1984[36]/1985 [III-72] extended the performance of motion-compensated interframe hybrid coding down to 50Kbits/s. In this work, motion estimation was compared using modified versions of Jain and Jain's logarithmic search and Ohtsuka and Ninomiya's iterative search algorithms, in which they were extended to sub-pixel accuracy (½-pixel). Each block was coded with a flag indicating whether the motion vector was zero or not in the first case or whether the difference vector was zero or not in the second case. This was a forerunner to the MC/No-MC mode in H.261 and the macroblock coding modes in MPEG. The quantizer was uniform with a fixed dead zone, and a variable threshold. The threshold was adjusted to maintain a fixed bitrate per frame, so a rate buffer was not used. Temporal subsampling was used and compared at various rates. Prediction block sizes of 8x8 and 16x16 were compared to study the tradeoff between bits allocated for motion vector side information vs. coefficient quantization. The experiments found best results with the second motion estimation algorithm, 4:1 temporal subsampling and 8x8 prediction size. The author foresaw further improvement using frame interpolation. Ericsson published a further paper in 1985 [III-73] containing an extensive review and comparison between the two structures of hybrid coding with the transform outside or inside the DPCM loop. These are shown in Figure 27. The paper compared several spatio-temporal pixel-based predictors, and several field vs. frame block-based predictors. The latter included a mode that switched between field and frame prediction on a block-by-block basis, as used in MPEG interlaced prediction. Motion-compensated prediction with sub-pixel accuracy was found to be best. Two other papers in this timeframe reported similar algorithms. Natarajan 1984 [III-74] reported a 64Kbits/s hybrid coder with motion compensated prediction and frame interpolation. The scheme differed in using a simple zonal sampling with uniform quantization of the DCT coefficients. In 1985 Kummerfeldt [III-75] also reported hybrid coding with motion-compensated interpolation.

At this point, virtually all the major components of the H.261 algorithm were in place. Therefore, the major components of MPEG1 were also in place, with the exception of bi-directional prediction. However, at this time there were alternatives still being considered, including temporal subsampling, zonal coding, variable thresholding of coefficients and frame interpolation. Convergence was still ahead. A useful comparison of many of the different approaches to videoconferencing systems was provided by Rao [III-76]. Performance curves were included, as were various figures for system configurations or algorithm features. Helpfully, some of these figures were taken from PCS presentations that are no longer available.

[36] Reported at PCS 1984.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com



a) Structure I:
Transform domain prediction

b) Structure II:
space domain prediction

Figure 27. Two Structures for Hybrid Coding. (Ericsson, 1985)

The first commercial products for videoconferencing appeared around this time. Contribution #4 to the CCITT H.261 standard in November 1984 summarized the CLI (Compression Laboratories Inc.) and NEC products [III-77]. The CLI system comprised a hybrid coding scheme with DPCM in the transform domain, and VLC coding. The NEC system used adaptive 3D prediction with block-based motion-compensated prediction (8x16 block size), but pixel-by-pixel prediction mode decision. Prediction errors and motion vectors were entropy coded. The first commercial product to use motion-compensated hybrid coding was the PictureTel C-2000 (aka "MCT"), introduced in 1986. It used motion-compensated prediction (8x8 blocksize) with fractional accuracy (down to 1/256), an adaptive low pass filter applied to the MC prediction, 8x8 DCT, a uniform quantizer with deadzone, and VLC. Apart from the 8x8 motion blocksize, the main differences compared to H.261/MPEG were:

a) MV coding. The system used 8x8 transform coding of H and V components (i.e., a transform block corresponded to 64x64 pels) followed by a spatial perturbation (of +/- 0.625 pels) to correct coding errors in critical blocks.
b) Interpolated MV field. The motion vectors were interpolated for each pel with a trapezoidal window to get a smooth MV field.
c) The positions of the nonzero coefficients were quadtree-encoded (instead of run length-encoded)
d) Deblocking filter. Two pels on each side of the transform border were filtered.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

To avoid artifacts, the inverse of the deblocking filter was applied before the frame difference was transform-coded.
e) No intra mode. Instead, there was a heavy adaptive LP filter (5x5 box filter).

Frame Interpolation

In the early temporal DPCM systems, frames (or fields) were dropped or skipped when the buffer became full. See Haskell 1976 [III-16]. In these systems, the receiver interpolated the missing frames using a simple linear interpolation of pixels that were collocated in the transmitted frames. While such a technique worked well for stationary areas of the scene, it introduced motion blur in moving areas. The solution was using motion compensated interpolation. The principle is illustrated in Figure 28 and comprises estimating motion vectors between the transmitted frames, then using a scaled version of those motion vectors to calculate the pixel values of the skipped frame(s). This was first proposed by Jain and Jain 1979/1981 [III-51][III-52][37]. It should be noted that there is a secondary benefit to frame interpolation. It was known for a long time that temporal averaging was very effective for reducing random video noise. Prior to motion compensation however, this had an unacceptable blurring effect on moving areas. The averaging of motion compensated data from two frames reduces video noise considerably in the interpolated frame. (This benefit also applies to bi-directionally predicted data, as discussed in the following section).

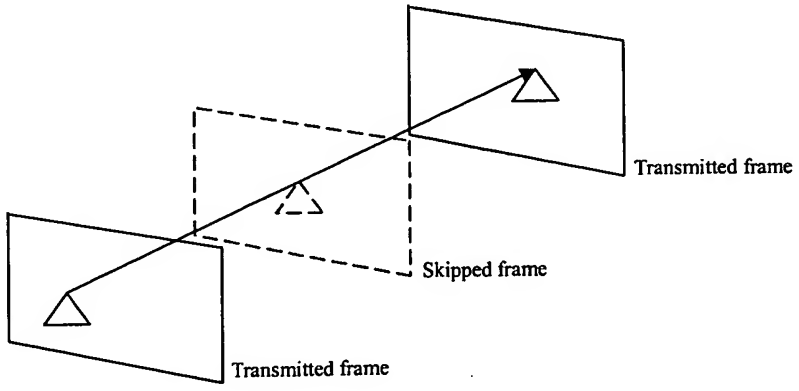


Figure 28. Frame Interpolation.

[37] D. Gabor showed a principle of motion interpolation in his paper in 1961. D. Gabor," Television Bandwidth Compression by Contour Coding," Proc.IEE, Paper No.3507E, pp.303-315, May 1961.

This approach was used by Lippmann in 1982[38] [III-78] for the RPV application. This was a rather special case because the movement was essentially deterministic, consisting of the forward movement of the aircraft, with the video camera typically pointing straight forward and down at an angle. The data was highly redundant because the speed of the aircraft was such that it would move only a fraction of a frame height in a frame time, therefore a given feature on the ground would appear in multiple frames. Simply skipping frames resulted in jerkiness that was disruptive to analysis of the data. It was not necessary to perform intensive motion estimation, owing to the deterministic nature of the motion, and selection of a few areas of the scene containing distinctive high-contrast features enabled accurate correlation between frames. The geometry of the flight could then be used to interpolate the missing frames.

Motion-compensated interpolation was applied to videoconferencing in 1984 by Furukawa, Koga and Iinuma [III-79], and by Bergmann [III-80][39]. The paper by Furukawa et al discussed the use of block-based motion estimation to calculate motion vectors between transmitted fields that could be scaled to provide motion-compensated prediction of skipped fields. The paper discussed the issue of erroneous motion vectors that are caused by noise or illumination changes and that caused discontinuities in moving objects (especially at boundaries of objects). They suggested a cost function based on the prediction error to eliminate false motion vectors. The effect was to make the motion vector field uniform, with a rigid-body assumption for the motion model. The technique was tested in real-time using the NETEC-X1MC motion-compensated codec[40].

Bergmann's technique segmented the scene into four areas – background, moving, uncovered, and to-be-covered. A different filter was used for each of the four areas. The to-be-covered area was determined by filtering the previous frame while the covered area was determined by filtering the future frame. A fast-converging PRA was used to estimate motion in the moving area, and the filter was the weighted sum of displaced pixels from the previous and future frames. This is similar in principle to the prediction modes of MPEG in which an area of the scene may be forward-predicted, backward-predicted or bi-directionally-predicted. The moving area filter used sub-pixel accuracy, and temporal-distance weighting for interpolation over multiple skipped frames.

Motion compensated field/frame interpolation was also developed for TV standards converters. In this application, fields or frames are created at temporal locations between original frames, so the application is not attempting to efficiently code the difference between a skipped or predicted frame. The accuracy of motion estimation is thus more critical because the motion vectors must represent true motion rather than just a good match. This situation can be illustrated by considering motion of an object consisting of

---

[38] Citations exist for papers on this topic being presented at PCS 1981 by both Bergmann and Netravali. These papers described PRA techniques for frame interpolation, with good results for small motion, but inaccuracy for higher motion. Unfortunately PCS does not publish proceedings, and no copy has been found of the abstracts typically handed out at these meetings.

[39] Bergmann earlier reported his work at PCS 1981.

[40] Note this was a spatial domain coder, not a hybrid coder.

repetitive texture, such as the pattern on clothing. In the coding application, matches may be found from one block to the next in blocks of the reference frame that contain the same fragment of the pattern, but which are not necessarily on the true motion path of the object. Such matches can result in small prediction error, and are thus suitable candidates for coding. For the standards conversion application however, such non-true motion vectors result in local perturbations of the object motion and are perceptually disturbing. This led to the development of techniques for estimation of true motion vectors, and more recently this has produced a benefit for coding efficiency. While the use of non-true motion vectors can produce satisfactory prediction of the pixel data, the differential coding of such motion vectors is less efficient because they are not uniformly aligned within a given object. Frame interpolation for standards conversion is described by Thomas 1987 [III-61]. This work also describes methods to find true motion vectors, the use of sub-pixel accuracy with higher-order spatial interpolators, and methods to deal with covered and uncovered background that will be discussed in the next section.

In October 1989, MPEG received fifteen proposals for the MPEG1 standard. Six of them included some form of motion-compensated frame interpolation. (A further three proposals included bi-directional prediction, which will be described in the next section). Four of these proposals simply skipped every other frame, and reproduced it with motion-compensated interpolation at the decoder. One proposal skipped two frames and the other proposal had a more complex combination of predicted and interpolated frames. All except one transmitted the prediction error for the interpolated frames. One proposal calculated separate motion vectors for the interpolated frames, by minimizing the prediction error from both the adjacent frames. The other proposals simply used the motion vectors from prediction of the adjacent frames.

Bi-Directional Prediction

Motion-compensated frame interpolation uses a linear, constant velocity model for motion. Further, it assumes that the integrity of objects is preserved within the scene. While this is an excellent approximation, the model fails some of the time – objects do not travel in straight lines and they do pass in front of each other. In particular experimenters developing frame interpolation methods reported artifacts and excessive noise in uncovered regions[41][42]. Bi-directional prediction solves these problems by performing motion estimation twice and generating an accurate backwards motion vector from the future frame to the interpolated frame and an accurate forwards motion vector to the interpolated frame as opposed to dividing the forward motion vector between the past and future frames. This provides a piecewise continuous model of motion that can accommodate lateral acceleration, and cope with occlusion and uncovering. See Figure 29. Backward motion compensated prediction was not considered for video coding until

[41] A comparison of SNR for interpolative coding vs. bi-directional coding shows very little difference. The perceptual difference however is larger, due to artifacts on moving edges and at uncovered regions where the eye is probably focusing on important information.
[42] Dynamic noise of this kind is often referred to a "mosquito" noise. The block artifacts twinkle in a manner reminiscent of mosquitoes flying around a lamp. It is also known as the "corona" effect.

MPEG1 with its requirements for "trick modes". Non-linear motion was not a strong issue for videoconferencing content, and researchers initially sought to solve the uncovered background problem by intraframe prediction or filtering. (See Bergmann 1984 [III-80]. However, Thomas suggested the use of backward motion vectors for uncovered background (and the use of forward motion vectors for covered background) in his 1987 report on standards conversion [III-61]. A further disadvantage of backward prediction is the requirement for buffering a set of frames at the encoder (and decoder) such that the future frame can be used to predict the current frame in a causal operation. This implies a significant delay in encoding which would be intolerable in a real-time conversational environment such as videoconferencing, and it requires significant amounts of memory that were unaffordable before the 1990s. However, MPEG1's requirements for fast-forward, skipping, and reverse play coupled with indifference to coder delay, and availability of inexpensive memory caused a rethinking of the basic data structure and organization of the encoding/decoding process. The requirements necessitated a regular series of intra pictures such that entry into the bitstream could be made. This also provided support for fast-forward operation. For reverse play, it was assumed that decoders could decode all the frames between two intra pictures and display them in reverse order. Three of the MPEG1 proposals combined these features with bi-directional prediction to provide both flexibility and high quality.

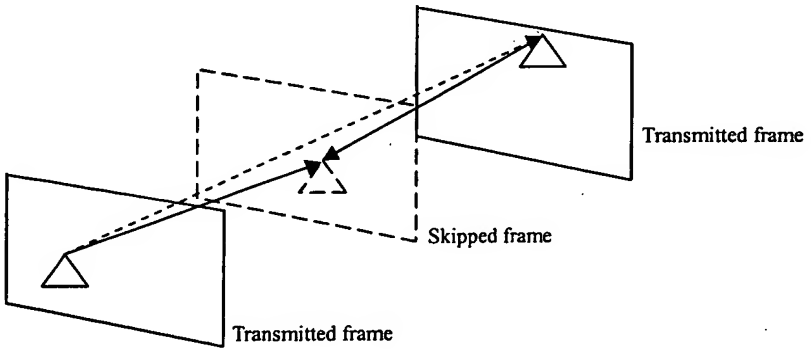


Figure 29. Bi-directional Prediction.

Yamada of JVC showed backward prediction and hinted at bi-directional prediction in a series of contributions to MPEG in 1989 that showed demo tapes and provided a terse description. See for example [III-81] and [III-82]. JVC's proposal to MPEG1 1989 [III-83] contains a full description of bi-directional coding and the use of four modes for coding a macroblock: intra, forward-predicted, backward-predicted and bi-directionally-predicted. Intra mode would be used when prediction failed. Forward prediction would be used when the object was occluded in the future frame. Backward prediction would be used when the object was uncovered in the future frame. Bi-directional prediction would be used in the normal situation when the motion is simple translation without occlusion. JVC proposed a fixed structure with Intra-pictures every six frames, and five B-pictures in between. There were no P-pictures.

Bi-directional prediction was also proposed by C-Cube [III-84], and Sony [III-85]. The former scheme was similar to the proposal of JVC but the number of B-pictures was variable. It seems from the terse description that only one of the forward or backward motion vectors was used per macroblock. The coding was done with three modes – "simple replenishment" with the motion-compensated block, coding of the prediction error, intra coding. Sony proposed a fixed structure with Intra-pictures every six frames (called a Group of Frames – GOF), but with a 2-level bi-directional prediction in which the center frame was bi-directionally predicted from the Intra-pictures, and then used together with one or other of the Intra-pictures to bi-directionally predict the two frames on either side of it.

During the development of the MPEG1 standard in 1990 in the convergence process after the tests, there was considerable discussion about fixing the format for the number of B-pictures and the format for the GOP. Initially many people felt these parameters needed to be fixed. To a considerable degree this was because it was thought that the amount of buffer memory in the decoder would depend on these parameters. (Essentially it was thought that the number of B-pictures would determine the number of frame buffers). It was also thought important to fix the format such that hardware state machine controllers could manage the decode process. In fact, a number of people (including the author) showed that the amount of buffer storage required was independent of the number of B-pictures, and in practice virtually all decoder designs used a software control strategy with a microcontroller instead of a state machine. Such an approach made complex control easy to implement. Therefore the standard permitted any number of B-pictures, and any arrangement of P-pictures within a GOP of arbitrary length. This is a particular example of the way in which MPEG distilled the standardization process to fix only the absolute minimum set of requirements. This left maximum flexibility for encoders to optimize performance, and for implementers to optimize cost.

Section III References

1. Y. Taki, M Hatori, S Tanaka, "Measurements on Interframe Correlation and Frame Difference Distribution of Television Signals", NHK Technical Report, Vol.11, No.3, pp.117-128, May 1959. (In Japanese)

2. C C Cutler "Differential Quantization of Communication Signals", US Patent 2,605, 631, Jul 1952.

3. B M Oliver, "Efficient Coding", BSTJ Vol. 31, No. 4, pp. 724-750, Jul 1952.

4. C W Harrison, "Experiments with Linear Prediction in Television", BSTJ Vol. 31, No. 4, pp. 764-783, Jul 1952.

5. R E Graham, "Predictive Quantization of Television Signals", IRE Wescon Convention Record, Part 4, pp. 100-106, Aug 1958.

6. J B O'Neal, "Predictive Quantizing Systems (Differential Pulse-Code Modulation) for the Transmission of television Signals. BSTJ Vol. 45, pp. 689-722, Oct 1966.

7. D J Connor, R F W Pease, W G Scholes, "Television Coding using Two-Dimensional Spatial Prediction", BSTJ Vol. 50, No. 3, pp. 1049-1061, Mar 1971.

8. A J Seyler, "The Coding of Visual Signals to Reduce Channel Capacity Requirements", IEE Monograph No. 535 E, pp. 676-684, Jul 1962.

9. A J Seyler, "Real-Time Recording of Television Frame Difference Areas", Proc. IEEE, pp. 478-480, Mar 1963.

10. A J Seyler, "Statistics of Television Frame Differences", Proc. IEEE, pp. 2127-2128, Dec 1965.

11. A J Seyler, Z L Budrikis, "Detail Perception after Scene Changes in Television Image Presentations", IEEE Trans on Info. Theory, pp. 31-43, Jan 1965.

12. F W Mounts, "A Video Encoding System with Conditional Picture-Element Replenishment", BSTJ, pp. 2545-2554, Sep 1969.

13. J O Limb, R F W Pease, "A simple Interframe Coder for Video Telephony", BSTJ Vol. 50, pp. 1877-1888, Jul-Aug 1971.

14. J C Candy, M A Franke, B G Haskell, F W Mounts, "Transmitting Television as Clusters of Frame-to-Frame Differences", BSTJ Vol. 50, No. 6, Jul-Aug 1971.

15. B G Haskell, P L Gordon, "Source Coding of Television Signals using Interframe Techniques", SPIE Vol. 66, pp. 9-22, 1975.

16. B G Haskell, "Interframe Coding of Monochrome Television – A Review", SPIE Vol. 87, pp. 212-221, 1976.

17. R F W Pease, J O Limb, "Exchange of Spatial and Temporal Resolution in Television Coding", BSTJ Vol. 50, pp. 191-200, Jan 1971.

18. T Ishiguro, K Iinuma, Y Iijima, T Koga, H Kaneko, "NETEC System: Interframe Coder for NTSC Color Television Systems", Third Intl Conf on Satellite Comm., Kyoto, Japan, pp. 309-314, Nov 1975.

19. T Ishiguro, K Iinuma, T Koga, S Azami, T Mune, "Composite Interframe Coding of NTSC Color Television Signals", 1976 NTC Record, pp. 6.4-1 – 6.4-5, Nov 1976.

20. H Yasuda, H Kuroda, H Kawanishi, F Kanaya, H Hashimoto, "Transmitting 4MHz TV Signals by Combinational Difference Coding", IEEE Trans. on Comm., Vol. COM-25, pp. 508-516, May 1977.

21. H Kuroda, N Mukawa, T Matsuoka, S Okubo, "1.5Mbits/s Interframe Codec for Videoconference Signals", Proc. GLOBECOM 82, pp. 987-991, 1982.

22. J E Thomson, "European Collaboration on Picture Coding Research for 2 Mbit/s Transmission", IEEE Trans. on Comm., Vol. COM-29, No. 12, pp. 2003-2004, Dec 1981.

23. J E Thomson, "Objectives and Results of Project COST 211", Proc. GLOBECOM 82, pp. 802-804, 1982.

24. T S Duffy, R C Nicol, "A Codec for International Visual Teleconferencing", Proc. GLOBECOM 82, pp. 992-997, 1982.

25. L Stenger, Th Kremers, R Govaerts, "Optimization of Coding Algorithms by Computer Simulation Studies", Proc. GLOBECOM 82, pp. 805-809, 1982.

26. L Chiariglione, R C Nicol, P Schaefer, "The Development of the European Videoteleconference Codec", Proc. GLOBECOM 82, pp. 810-814, 1982.

27. D I Crawford, "Spatio/Temporal Prefiltering for a Videoconference Codec", Int. Conf. on Electronic Image Proc., York, pp. 236-242, Jul 1982.

28. Nordquist.

29. T. Koga, et al, "Motion-compensated adaptive intra-interframe prediction and entropy coding," The Journal of Institute of Television Engineers of Japan, Vol.39, No.10, pp.963-971, Oct. 1985.

30. N. Mukawa and H. Kuroda, "Uncovered Background Prediction in Interframe coding," IEEE Trans. on Comm. Vol.COM-33, No.11, pp.1227-1231, Nov 1985.

31. M R Schroeder, US Patent 3,679,821. Jul 1972.

32. C Reader, "Orthogonal Transform Coding of Still and Moving Pictures", Ph.D. Thesis, School of Applied Science, U Sussex, England, 1973.

33. A Habibi, "Hybrid Coding of Pictorial Data", IEEE Trans. on Comm., Vol. 22, No. 5, pp. 614-624, May 1974.

34. H Whitehouse, E Wrench, A Weber, G Claffie, J Richards, W Schaming, J Schanne, " A Digital Real Time Intraframe Video Bandwidth Compression System", SPIE Vol. 119, pp. 64-78, 1977.

35. J A Roese, G S Robinson, "Combined Spatial and Temporal Coding of Digital Image Sequences", SPIE Vol. 66, pp. 172-180, 1975.

36. J A Roese, W K Pratt, G S Robinson, "Interframe Cosine Transform Image Coding", IEEE Trans. on Comm., Vol. COM-25, No. 11, pp. 1329-1339, 1977.

37. H W Jones, "A Conditional Replenishment Hadamard Video Compressor", SPIE Vol. 119, pp. 91-98, 1977.

38. H W Jones, "A Comparison of Theoretical and Experimental Video Compression Designs", IEEE Trans. on Electromag. Compat. Vol. EMC-21, No. 1, pp. 50-56, Feb 1979.

39. H W Jones, Comparison of Video Fields and Frames for Transform Compression", SPIE Vol. 149, pp. 214-221, 1978.

40. F Rocca, "Television Bandwidth Compression Utilizing Frame-to-Frame Correlation and Movement Compensation", Symposium on Picture Bandwidth Compression, MIT 1969, T S Huang, O J Tretiak, (Eds.), Gordon and Breach, 1972.

41. F Rocca, S Zanoletti, "Bandwidth Reduction via Movement Compensation on a Model of the Random Video Process", IEEE Trans. on Comm., pp. 960-965, Oct 1972.

42. C Cafforio, F Rocca, "Methods for Measuring Small Displacements of Television Images", IEEE Trans. on Info. Theory, Vol. IT-22, No. 5, pp. 573-579, Sep 1976.

43. S Brofferio, F Rocca, "Interframe Redundancy Reduction of Video Signals Generated by Translating Objects", IEEE Trans. on Comm., Vol. COM-25, pp. 448-455, Apr 1977.

44. C Cafforio, F Rocca, "Tracking Moving Objects in Television Images", Signal Processing Vol. 1, No. 2, pp. 133-140, Apr 1979.

45. J O Limb, J A Murphy, "Measuring the Speed of Moving Objects from Television Signals", IEEE Trans. on Comm., p. 474-478, Apr 1975.

46. J O Limb, J A Murphy, "Estimating the Velocity of Moving Images in Television Signals", Computer Graphics and Image Processing, Academic Press, pp. 311-327, 1975.

47. A N Netravali, J D Robbins, "Motion-Compensated Television Coding: Part 1", BSTJ, Vol. 58, No. 3, Mar 1979.

48. Y Taki, M Hatori, S Tanaka, "Interframe coding that follows the motion", Proceedings of 1974 IECEJ (Institute of Electronics and Communication Engineers of Japan) Annual Convention, p. 1263, July 1974.

49. S Brofferio, C Cafforio, P Del Re, G Quaglia, A Racciu, F Rocca, "Redundancy Reduction of Video Signals Using Movement Compensation", Alta Frequenza, pp. 836-843, Oct 1974.

50. F Giorda, A Racciu, "Bandwidth Reduction of Video Signals via Shift Vector Transmission", IEEE Trans. on Comm., pp. 1002-1004, Sep 1975.

51. J R Jain, A K Jain, "Interframe Adaptive Data Compression Techniques for Images", Signal and Image Processing Lab., Dept. Elec. Eng., Univ. California, Davis. Aug 1979.

52. J R Jain, A K Jain, "Displacement Measurement and its Application in Interframe Image Coding", IEEE Trans. on Comm., Vol. COM-29, No. 12, pp. 1799-1808, Dec 1981.

53. T Koga, K Iinuma, A Hirano, Y Iijima, T Ishiguro, "Motion-Compensated Interframe Coding for Video Conferencing", Proc. NTC, pp. G5.3.1-G 5.3.5, Nov 1981.

54. Y Ninomiya, Y Ohtsuka, "A Motion-Compensated Interframe Coding Scheme for Television Pictures", IEEE Trans. on Comm., Vol. COM-30, No. 1, Jan 1982.

55. S Kappagantula, K R Rao, "Motion Compensated Predictive Coding", SPIE, Vol. 432, pp.64-70, 1983.

56. R Srinivasin and R Rao, "Predictive Coding Based on Efficient Motion Estimation", Proc. ICC, pp. 521-526, May 1984.

57. Bierling.

58. J A Stuller, A N Netravali, "Transform Domain Motion Estimation", BSTJ, Vol. 58, No. 7, Sep 1979.

59. H C Bergmann, "Displacement Estimation Based on the Correlation of Image Segments", Int. Conf. on Electronic Image Proc., York, pp. 215-219, July, 1982.

60. T R Hsing, "Motion Detection and Compensation Coding for Motion Video Coders: Technical Review and Comparison", pp. 2.6.1-2.6.5, 1987.

61. G A Thomas, "Television Motion Measurement for DATV and Other Applications", BBC Research Dept. Rep., 1987/11, Sep 1987.

62. D R Walker, K R Rao, "Improved Pel-Recursive Motion Compensation", IEEE GLOBECOM'84, pp. 1128-1134, Oct 1984.

63. J Biemond, L Looijenga, D E Boekee, "A Pel-Recursive Wiener-Based Displacement Estimation Algorithm for Interframe Image Coding Applications", SPIE, Vol. 845, pp. 424-431, 1987.

64. K Iinuma, Y Iijima, T Ishiguro, H Kaneko, "Interframe Coding for 4 MHz Color Television Signals", ICC 75, pp. 23-26 – 23-30, 1975.

65. T Koga, A Hirano, K Iinuma, Y Iijima, T Ishiguro, "A 1.5Mbit/s Interframe Coder with Motion Compensation", Proc. ICC83, Vol. 2, pp. 1161-1165, 1983.

66. T Koga, K Iinuma, K Niwa, S Tsugane, M Nishiwaki, Y Iijima, "Sub-T1 Rate Motion Video Codec for Teleconference", Proc. ICC85, Vol. 1, pp. 2.4.1-2.4.5, 1985.

67. K Iinuma, T Koga, K Niwa, Y Iijima, "A Motion-Compensated Interframe Codec", SPIE, Vol. 594, pp. 194-200, 1985.

68. Y Ninomiya, Y Ohtsuka, "A Motion-Compensated Interframe Coding Scheme for NTSC Color Television Pictures", IEEE Trans. on Comm., Vol. COM-32, No. 3, Mar 1984.

69. F J S Rivero, "Multimode Predictive Coding Algorithm with Motion Compensation", SPIE Vol. 594, pp. 111-118, 1985.

70. A N Netravali, J A Stuller, "Motion-Compensated Transform Coding", BSTJ, Vol. 58, No. 7, Sep 1979.

71. A Furukawa, T Koga, K Niwa, "Coding Efficiency Analysis for Motion-Compensated Interframe DPCM with Transform Coding", Proc. Globecom85, Vol. 2, pp. 689- 693, 22.5.1-22.5.5, 1985.

72. S Ericsson, "Fixed and Adaptive Predictors for Hybrid Predictive/Transform Coding", IEEE Trans. on Comm. Vol. COM-33, No. 12, pp.1291-1302, Dec 1985.

73. S Ericsson, "Motion-Compensated Hybrid Coding at 50 Kb/s", Proc. ICASSP, pp. 10.8.1-10.8.4, Mar 1985.

74. R Natarajan, K R Rao, "Design of a 64 KBPS Coder for Teleconferencing", SPIE Vol. 504, pp. 406-413, 1984.

75. G Kummerfeldt, F May, W Wolfe, "Coding television Signals at 320 and 64 Kbits/s", SPEI Vol. 594, pp. 119-128, 1985.

76. K R Rao, Digital Signal Processing for Low Bit Rate Image Codecs", SPIE Vol. 757, pp. 8-15, 1987.

77. USA, "Description of two of the Video Codecs used in the United States with Primary Rate Transmission Systems", Contribution #4, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Nov 1984.

78. R Lippmann, "Continuous Movement Regeneration in Low-Frame-Rate Aerial Images", Int. Conf. on Electronic Image Proc., York, pp. 194-198, July, 1982.

79. A Furukawa, T Koga, K Iinuma, "Motion-Adaptive Interpolation for Videoconference Pictures", Proc. ICC84, pp. 707-710, May 1984.

80. H C Bergmann, "Motion Adaptive Frame Interpolation", Zurich, 1984

81. Y Yamada, "Beach and Flower Garden Demonstration", ISO/IEC JTC1/SC2/WG8 MPEG 89/005, Feb 1989.

82. Y Yamada, K Sugiyama, "Simulation of Picture Coding Method 'BANC' ver .3", ISO/IEC JTC1/SC2/WG8 MPEG 89/105, May 1989.

83. T Hidaka, "Description of the Proposing Algorithm and its Score for Moving Image (A Part of the Proposal Package)", ", ISO/IEC JTC1/SC2/WG8 MPEG 89/188, Oct 1989.

84. A Ligtenberg, "No Bells", ISO/IEC JTC1/SC2/WG8 MPEG 89/183, Oct 1989.

85. Sony "Description of the Proposed Coding Algorithm", ISO/IEC JTC1/SC2/WG8 MPEG 89/194, Oct 1989.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

Section IV Early Coding Standards

H.120

To be written.

H.261

Work on the H.261 Standard began in 1984 in a Specialists Group of CCITT SG XV (Study Group Fifteen). It inherited the technology of H.120, and part of the H.120 workplan that was still to be completed[43]. In addition to Part 3 of H.120, the mandate included two work items – coding at n x 384 Kbits/s and coding at m x 64 Kbits/s. Initially the latter two items were seen as distinct and presumably requiring different coding algorithm technology. The work on n x 384 was begun first, with serious consideration of the m x 64 work item beginning in 1988. [44]The H.261 committee was constituted by many of the same people as H.120, and was influenced very strongly by the COST 211 project. Therefore the natural starting point for H.261 was adaptive 3D prediction, with motion compensation as an option. (In summary: spatial, i.e., intraframe prediction for stationary areas and fast-moving areas, motion compensated prediction for slowly moving areas, and background prediction for uncovered areas). However, the COST 211 committee had been impressed by a demonstration of hybrid (DCT-DPCM) coding at a meeting in 1984 at Martlesham, England[45], and the H.261 work quickly evolved into a strategy of predictive coding plus other coding, in which the other coding was either DCT-based or VQ-based (Vector Quantization). At the third meeting in September 1985 [IV-1], the committee reviewed a number of proposed algorithms, including pixel-domain predictive, hybrid DCT, hybrid VQ and pyramid coding schemes. The committee decided to focus on "interframe prediction with further processing", but at this stage the further processing could be inside or outside the prediction loop. A key proposal for hybrid DCT coding was provided by six European countries 1985 [IV-2][46], and featured three prediction modes – Intra, Inter, and Inter plus motion compensated. Blocks could be skipped ("conditional replenishment"), and higher frequency DCT coefficients could be dropped ("frequency weighting"). Adaptive quantization was provided to take account of non-stationarity. The importance of side information was recognized in a Japanese contribution that reviewed three of the candidate algorithms including hybrid coding 1985 [IV-3]. Side information included "significant/insignificant block" flag, i.e., coded/not-coded flag, motion vector and

---

[43] H.120 Part 3. Coding for 525 lines, 60 fields/s, 1544Kbits/s. Essentially an NTSC version of H.120. This included motion-compensated interframe coding.

[44] See H.261 contribution #346 the 12th Meeting report Mar 88 [III-120], for the first discussion of px64. The letter "p" denoted the merger of nx384 and mx64. By the meeting in September 88 the Reference Model had become the basis for 64kbps coding and there was almost convergence. It was agreed the development of px64 would be on an accelerated schedule.

[45] Hosted by British Telecom Research Labs.

[46] In general, contributions to the H.261 Specialists Group were attributed to countries, rather than organizations or individuals.

quantization set index. The third meeting also defined the CIF (Common Interchange Format), initially with 360 pixels/line. This was a brave compromise between the PAL and NTSC formats that was based on the PAL spatial format and NTSC temporal format.

Following the meeting, a series of contributions explored the proposed algorithms. The group of Japanese companies proposed a generic structure for the codec 1986 [IV-4] and a coding strategy for the DCT-based approach that focused on a mechanism to deal with troublesome large magnitude DCT coefficients not fitting the statistical model 1986 [IV-5]. Two contributions in 1986 proposed the use of frame interpolation [IV-6][IV-7] (See the next section of this paper), and a further contribution considered both frame interpolation and object-based motion compensated prediction 1986 [IV-8]. This paper raised a concern to be heard frequently – the cost of transmitting motion vector and other side information. It concluded that block-based motion was better than object-based, but this may have reflected a lack of homogeneity in the motion vector field. A number of contributions over the next two years commented on this problem, which was exacerbated by both the nature of videophone content and the low resolution of the videophone format. All these papers presaged work to be done later in MPEG. A contribution from Sweden, Netherlands, France and Federal Republic of Germany (FRG) 1986 [IV-9] is notable for proposing different modes on a frame basis. This included intra-frames as used by MPEG, although these did not end up in the final H.261 specification. The first proposal for the video multiplex (i.e., syntax) came from France and Italy in contribution #69 1986 [IV-10]. This introduced the concept of "attributes" that included modes of operation such as Intra/Non-Intra, and parameters such as frame rate. Figure 30 illustrates the proposal.

- Block Transport Attributes:
  - Fixed[47] /Non-Fixed
  - Motion Compensated/Not-Motion Compensated
  - Coded/Not-Coded
  - Intra/Non-Intra
- Block Presentation Attributes:
  - Class Number
  - Last Coefficient Address
  - Quantizer Number
- Field Attributes:
  - Split-screen
  - Buffer Size
  - Subsampling Mode

Figure 30. H.261 Contribution #69. (France and Italy, 1986).

[47] H.261 used the ambiguous term "fixed" to denote blocks that were not moving. Since motion compensation was optional, a moving block could be predicted without motion compensation. Hence it was necessary to signal moving blocks separately.

It should be noted that the concept of attributes was already familiar to CCITT (ITU-T) in the context of communication services. The paper cites I-series recommendations Red Book I 130 and I 211.

At the 4th meeting 1986 [IV-11] the CIF format was confirmed, and 3D pixel-domain prediction was dropped – only "interframe prediction + further coding" remained. Motion compensated interpolation remained an option. The video mux was advanced significantly in a contribution from UK, Sweden and France 1986 [IV-12]. This introduced the concept of a hierarchical structure (Field, Group-of-Blocks and Block) and included the following key elements for the first time:

- Start codes (not yet in the familiar final form)
- Buffer state
- Temporal reference (for the field)
- Field type – a mode to be used for all blocks in the field (e.g., intra)
- Block line number (equivalent to Slice Vertical Position in MPEG)[48]
- Block address, VLC coded relative to the start of the line or the previous coded block (equivalent to macroblock address in MPEG)
- Block type, VLC coded attributes (cf. Contribution #69)

Note however that luminance and chrominance blocks were "TDM" multiplexed on a line-of-blocks basis, with two rows of luminance blocks first, followed by a row of the U chrominance blocks, then a row of the V chrominance blocks. This is shown in Figure 29. Contribution #88 from FRG considered several approaches to efficiently code the motion vectors 1986 [IV-13]. Fixed wordlength, variable wordlength and variable wordlength of differences were considered, with citation to several papers discussed earlier (e.g., Jain and Jain 1981 [III-52], Kummerfeld et al 1985 [III-75]). This is the first of a series of contributions that addressed this issue, wrestling with the homogeneity of the motion vector field, panning vs. object-based motion, and singly vs. jointly (i.e. 2D) coding the two components. Contribution #90 from Sweden 1987 [IV-14] tackled issues of intra and inter quantization with and without a dead zone, joint coding of attributes, and the use of multiple scan patterns as opposed to a single zigzag scan. This was a precedent to the alternate scan used in MPEG2. The paper touched briefly on buffering issues. Contribution #94 from France considered the use of variable blocksize DCTs 1986 [IV-15]. Contribution #102 1986 [IV-16] was a continuation of #85 for the video mux, and contained examples for VLC tables and revised start codes.

[48] A similar concept was used in H.120 Ver. 1.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

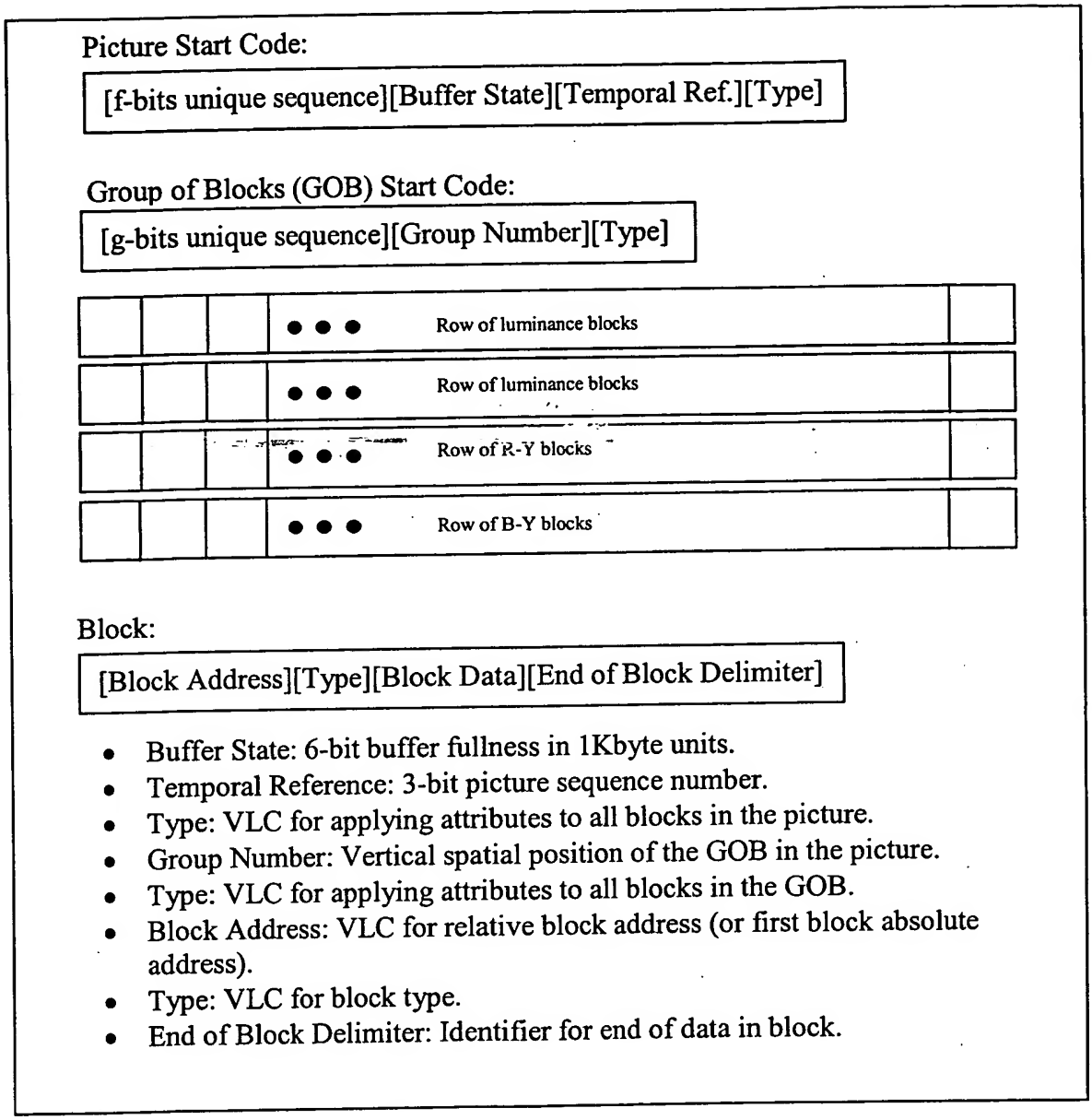


Figure 31. H.261 Contribution #82. (UK et al, 1986).

At the Fifth meeting in March 1986, VQ was dropped, and the blocksize was fixed at 8x8 pixels for both luminance and chrominance 1986 [IV-17]. A reference model (RM1) was defined – see Annex 2 and document #104 1986 [IV-18]. The principal of the reference model was established by the hardware developed for field trials in H.120. At this point many of the alternatives were dropped, the basic algorithm was set, and many of the small coding tools were in place. From this point on, the committee pursued a set of key

areas of technology for standardization. These included the following video-coding related items:

- Picture format.
- Motion vector coding.
- Coefficient coding including scanning patterns and thresholding.
- Coding of block attributes, i.e., "side information".
- Coding of block addresses
- Quantization.
- Bitstream format, called "Video Mux".
- Filtering – Pre-processing, Post-processing and especially Loop filtering and the tradeoff against sub-pixel motion compensated prediction.
- IDCT mismatch
- Buffering

In addition, a number of other topics addressed system-level issues such as framing, call setup and control, and timing/sync.

In each of the areas, extensive investigation was carried out to optimize the algorithm. Many options were considered, simulated, and traded off against each other and implementation complexity. As such, far more coding tools were examined than ended up in the final H.261 standard, but some of the innovations from this period did end up in the MPEG standard. As noted, the alternate scan of MPEG2 was preceded by investigation of multiple scan patterns in H.261. Even object-based coding as used in MPEG4 was first examined in H.261! The details of every twist and turn of the development is not relevant or necessary here. The following summary highlights specific tools that were first introduced during H.261 development that later became part of MPEG. It should be emphasized again that the analysis and tradeoffs performed during H.261 were specific to the type of data in videoconferencing and to the very low bitrates used. Some of the techniques were dedicated to saving 3-4 bits per coded block or even only 1 bit. Such savings amounted to 3-5% at 64 Kbits/s, but are negligible at MPEG1 rates, let alone MPEG2 rates. Nevertheless, such techniques were preserved in MPEG in pursuit of backward compatibility. Unfortunately they may encumber MPEG with needless complexity. The developments in each area are summarized in the following paragraphs.

The CIF picture format was modified slightly to be 352 pixels/line 1986 [IV-17]. A smaller picture format for 64Kbits/s was proposed in contribution #279 1986 [IV-19]. This became known as QCIF. There was also a proposal for an intermediate format called 4/9CIF, but this was not adopted.

The VLC coding of differential motion vectors was further considered in contributions from Japan 1986 [IV-20] (also published in IEEE [IV-21]) and the UK 1987 [IV-22]. Both examined using 2D VLCs to code the vertical and horizontal components of the motion vector. This is an example of the previously established technique of joint probability coding, and is similar to the use of 2D VLCs for run-length coding of the transform coefficients. The former contribution also introduced (and the latter

contribution clarified) the trick for coding differential motion vectors in which the limited range of the motion vectors was used to correctly interpret the differential code[49]. The values could be coded with fewer bits as a result. Further simulations showed that 1D VLC coding of differential motion vectors with a single table was almost as efficient as 2D coding or adapting the table, so this was adopted for simplicity. See contribution #416 1988 [IV-23]. It was also observed that there was no penalty in extending the range to +/- 15 pixels when VLC coding was used. Note this contribution also reported that 8x8 motion vectors were slightly better, but this was not adopted.

Defining a set of VLCs for coefficients was addressed by contribution #122, in which it was noted that the EOB code was statistically very frequent and should be a member of the VLC set 1986 [IV-24]. Further contributions studied the use of multiple VLC tables vs. a single table for both coefficients and attributes, and concluded there was little or no gain from multiple tables. This is another example wherein the sensitivity of the algorithm to data statistics is too low to merit the use of measured statistics to perform local optimization. Contribution #152 recommended the use of a fixed 9-bit prefix for codes greater than 10 bits in length, i.e., an escape code mechanism 1986 [IV-25][50]. In November 1986, a key contribution from the Federal Republic of Germany, #170, introduced the use of 2D VLCs for coding the run-length of zeros and level of the non-zero coefficient terminating the run [IV-26]. The paper reported the technique was most useful at low bitrates, where the saving in bitrate was 11-20% on two test sequences. Note however that this was only 3-4 bits per block. The technique is an example of coding with joint statistics, which was a known approach, but not previously applied to video coding[51]. A Japanese contribution, #188 confirmed the findings, and reported 3.5-10% improvement in bitrate for fixed SNR 1987 [IV-27]. The next major advance in this area was reported in contribution #234 from FRG, France, GEC, UK and SEPA 1987 [IV-28]. This was the introduction of macroblocks, which could be viewed from several different perspectives. On the one hand, it was originally presented as a common format to be used for both 384Kbits/s coding and 64 Kbits/s coding. On the other hand, this was a multiplex at a finer level of granularity than the previous method of multiplexing lines of blocks of luminance and chrominance data. It was further presented as a coding efficiency in which one motion vector and other attributes could be shared among four luminance blocks, thereby saving overhead bits. The contribution also proposed the differential coding of DC-terms within the macroblock. Macroblocks are discussed further below in the context of the video multiplex. Contribution #435 studied the performance of 2D VLCs vs. 2, 1D VLCs 1988 [IV-29]. It concluded that the performance and the implementation cost were about the same. The addition of a stuffing codeword to avoid buffer underflow was introduced in contribution #387 from FRG [IV-30]. The set of VLCs was modified slightly to avoid start code emulation. More analysis of DC-term prediction was made in two contributions concerned with the intra coding of

[49] This is accomplished with a 2's complement representation.
[50] Note this is also relevant to "fast" decoder implementations that use some kind of dual table approach with parallel decoding.
[51] The use was also claimed by CLI in the same timeframe. It has also been claimed that the technique was used earlier in proprietary videoconferencing equipment.

the frame after a scene change 1988 [IV-31] 1989 [IV-32]. Essentially the scheme comprised transmitting only the DC terms differentially in an "Intra-not-coded" mode as a way of avoiding complete buffer overload after only a few normally coded intra blocks.

The coding of block attributes considered the use of a single VLC table or multiple VLC tables for the different attributes and also for the differential block address. Studies such as #231 1987 [IV-33] found that the variation between different test sequences was large enough that optimization for each sequence was not feasible. It also found that a table commencing with a 1-bit code was significantly better than one starting with a 2-bit code, and concluded that at most two tables like these were necessary[52]. Contribution #357 introduced a new attribute as a consequence of the introduction of macroblocks 1988 [IV-34]. Prior to this, it was necessary to send the EOB code for every block, even if the block was not coded. Coded Block pattern – CBP – was a VLC for the macroblock signaling which blocks were coded. Originally seven sets of patterns for coded blocks were considered with various numbers of patterns in each set. The full set of 63 patterns was chosen in contribution # 449 1989 [IV-35], and confirmed in contribution #461 1989 [IV-36]. This contribution also introduced the so-called first-coefficient trick for the coefficient VLCs[53]. Since the first coefficient could not be EOB, a slightly different, more efficient VLC table that did not include the VLC code for EOB could be used. This is an excellent example of the type of extreme optimization used in H.261 to produce the minimum number of bits for very low bitrate coding.

The coding of block addresses was discussed in contribution #122 1986 [IV-24]. The use of an escape code mechanism to limit wordlength was introduced. The use of "relative" addresses to code runs of skipped blocks was discussed in contribution #146 1986 [IV-37]. After the introduction of macroblocks, contribution #449 further discussed the VLCs for macroblock address, including macroblock stuffing as a member of the set 1989 [IV-38].

Many contributions concerned the type of quantizer (e.g., uniform vs. non-uniform); whether or not to have a dead-zone, and the size of it; how many steps and the dynamic range; and how often in the multiplex to permit the stepsize to be changed. Some contributions produced directly conflicting results on these issues, probably highlighting again the relative insensitivity of the overall coding scheme to specific test sequences and conditions. A few highlights are contribution #149 that introduced the use of 32 steps 1986 [IV-39], contribution #542 that proposed the final range of 2-62 for the steps 1989 [IV-40], contribution #384 that proposed adjusting the step size on a GOB and block basis 1988 [IV-41], and contribution #480 which pointed out that adjustment at the macroblock level was essential for QCIF format 1989 [IV-42].

Initially the video multiplex (mux) was an adaptation of the simple video raster scan with lines of blocks, and luminance lines interleaved with chrominance lines of blocks 1986

[52] H.261 uses the same table for macroblock address and differential motion vectors.
[53] Attributed to a letter from Geoff Morrison, Feb 1 1989.

[IV-12]. It was recognized that attribute data should be multiplexed with each block to facilitate immediate decoding with the minimum buffering 1986 [IV-10]. The video mux evolved considerably with the introduction of macroblocks, the impact of which on coding efficiency was studied in contribution #266 1986 [IV-43]. As reported in the 7th Meeting Report, macroblocks were found to provide better performance at 64Kbits/s, but no improvement at 384Kbits/s[54] 1986 [IV-44]. The impact of macroblocks is therefore more related to implementation complexity than coding efficiency[55]. Macroblocks were included in RM5 1988 [IV-45] and became a permanent part of the multiplex. A single format for GOBs in both CIF and QCIF pictures was introduced in contribution #385 1988 [IV-46]. The issue of intra pictures was discussed several times, for example contribution #184 1987 [IV-47], but intra pictures were not adopted until the MPEG standard.

Many contributions discussed filtering. Pre-filters, post-filters and filters in the DPCM loop – "loop filters" – were analyzed extensively. Comparison was also made multiple times between sub-pixel motion compensated prediction and loop filtering. A comprehensive summary will not be given here. Rather a few highlights will illustrate the work. Contribution #207 1987 [IV-48] concluded that post-filtering has a greater impact on final quality than many of the fine-tunings to VLC tables etc. Contribution #300 extensively studied sub-pixel accuracy with several spatial interpolation filters 1988 [IV-49]. It concluded that ½-pixel accuracy was sufficient for videophone pictures and ¼-pixel accuracy was sufficient for broadcast TV applications. Contribution #416 concluded that the loop filter was as good as ½-pixel and far less complex to implement 1988 [IV-50]. Contribution #475 contains a comprehensive study of filters and their placement 1989 [IV-51].

From the first reference hardware model it was recognized that the accuracy of the IDCT must be such that computational error would not cause divergence in the DPCM loop at the decoder. At first, this was thought to be a matter of specification, see for example the 12th Meeting Report 1986 [IV-52]. A subcommittee was formed to investigate and established a process with chip manufacturers to establish a specification and conformance process. The IEEE agreed to establish a standard (P1180/D2). As time went on however, it was realized that mismatch could occur due to rounding errors even with implementations conforming to tight specifications. The problem was partly tackled by the technique known as "oddification" in the inverse quantizer. This was introduced by Japan in contribution #559 1989 [IV-53]. At the final meeting of H.261, IDCT mismatch was reported as the only problem with field trials of the completed algorithm. No real solution to mismatch was developed during H.261, but the committee reported the problem to ISO and IEEE, and agreed on oddification for quantization.

[54] These results are subject to the constraints of the videophone data type, and the homogeneity of the motion vector field.
[55] For example, it is easier to implement a decoder with minimum buffering when the blocks and their attributes are co-located kin the bitstream.

Clipping of the data output from the IDCT was a recognized issue. See contribution #143 1986 [IV-54].

Buffering of the variable-rate data for transmission and the attendant delay were the subject of a series of contributions. A solution was found that is the direct ancestor of the scheme used in MPEG. This was known as HRD – Hypothetical Reference Decoder. Its operation is described in the 16th Meeting Report 1989 [IV-55].

Several pieces of coding technology were not adopted by the standard. The use of multiple scan classes was studied for a time. The choice of up to 8 classes of scan was considered. Studies showed that alternate scans were not effective after 2D VLCs were introduced, and not effective at low bitrates. Object-based coding had the goal of allocating more bits to the facial region of the typical videophone picture. It was not shown to be effective at the time. The use of a background memory for prediction of uncovered areas was also investigated. Lastly, the use of variable block sizes for prediction and DCT coding was examined. These techniques were all used later in MPEG.

The Meeting reports provide a comprehensive summary of the H.261 work. Each report contains a summary of every contribution as well as enumeration of the various decisions made in each area of development. Status of the H.261 development was reported publicly in 1987 by the Chairman and several key contributors [IV-56]. The standardization process itself, and a summary of key milestones is contained in an excellent 1995 paper by the Chairman [IV-57].

CMTT

An effort was initiated in the late 1980s by CCIR to create a professional standard for compressed video. The application areas included contribution and distribution, and the bitrates envisaged ranged from 34Mbits/s to 140Mbits/s. Pixel-based adaptive prediction and hybrid coding were candidates. A high-level overview is given by Stenger 1990 [IV-58]. The effort, known as the CMTT coder did not progress well and was eclipsed by the MPEG2 effort.

MPEG1

The MPEG1 standard immediately followed the H.261 standard, and fundamentally comprises the same set of coding tools. The basic hybrid DCT/DPCM algorithm with 2D VLC coding is the same. The MPEG VBV rate buffer control is the same concept as the HRD in H.261. Everything else was either copied directly from H.261 or slightly modified. The differences can be summarized as extensions and greater flexibility to support the different applications targeted by MPEG1. While H.261 was focused on videoconferencing via traditional switched networks, MPEG1 was primarily focused on interactive video compact disk applications. As such, the video format needed to be flexible so that sub-windows could be provided, the range of content was far wider including film content and broadcast TV content such as sports, and support was needed

for the so-called "trick modes" of VCRs such as fast-forward. The differences between MPEG1 and H.261 are summarized below. Note that some of them concern techniques that were developed in H.261 but which did not become part of that standard.

MPEG1 additions and modifications to H.261:

Coding Tool	Comments
Longer Start Codes.	
GOP & Sequence layers.	ITU standards typically transmit this type of information "out-of-band" in channels described by other standards.
Flexible Picture format.	
Generalized Slice structure.	Replaced the rigid H.261 GOB structure.
Intra-Pictures.	H.261 has Macroblock forced updating, but experimented with intra-pictures.
Half-pixel interpolation.	This replaced the H.261 loop filter. Half-pixel interpolation was investigated within H.261 and was already adopted by CMTT before MPEG-1 reached its first draft syntax proposal.
Bi-directional prediction (B-Pictures).	
Extensive motion vector ranges.	H.261 is limited to +/- 15 pixels
Extended coefficient VLC tables.	
Slightly improved coefficient quantizer.	
Quantization weighting matrices.	From JPEG.

Only bi-directional prediction is a significant change. It is described in a separate section below. The other changes are summarized in the section on small coding tools.

Section IV References

1. Chairman of the Specialist Group on Coding for Visual Telephony, "Report of the Third Meeting in Torino (September 24-27 1985)", Contribution #54, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Sep 1985.

2. Sweden, Netherlands, France, Federal Republic of Germany, United Kingdom, Italy. "Hybrid Coding", Contribution #36, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Sep 1985.

3. KDD, NTT, NEC, Fujitsu, "Candidates for Subrate Coding Algorithm", Contribution #43, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Sep 1985.

4. KDD, NTT, NEC, Fujitsu, "Generic Structure of n x 384 Kbits/s Codec", Contribution #60, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Jan 1985 (sic).

5. KDD, NTT, NEC, Fujitsu, "DCT-Based Coding Algorithm", Contribution #61, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Jan 1986.

6. AT&T Bell Laboratories, "Coding Full Color Motion Video at 300-400 Kilobits/s Some Computer Simulation Results", Contribution #63, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Jan 1986.

7. FRG, "Motion Compensating Field Interpolation", Contribution #66, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Jan 1986.

8. FRG, "On Blocksize and Overhead for Displacement Vectors in Block Matching and Object matching Videotelephone Codecs", Contribution #67, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Jan 1986.

9. Sweden, Netherlands, France, Federal Republic of Germany, "Properties of DCT Based Hybrid Coder", Contribution #68, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Jan 1986.

10. France, Italy, "A Contribution to Video Multiplex Coding", Contribution #69, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Jan 1986.

11. Chairman of the Specialist Group on Coding for Visual Telephony, "Report of the Fourth Meeting in Ipswich (January 21-24 1986)", Contribution #74, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Sep 1985.

12. UK, Sweden, France, "Video Multiplex for nx384Kbits/s", Contribution #85, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1986.

13. FRG "Statistical Properties and Encoding of Displacement Vectors obtained by a Block-Based Method", Contribution #88, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1986.

14. Sweden, "Swedish Hybrid Coding Algorithm", Contribution #90, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1986.

15. France, "Variable Blocksize Hybrid Coding Scheme", Contribution #94, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1986.

16. UK, Sweden, France, "Video Multiplex for nx384Kbits/s", Contribution #102, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1986.

17. Chairman of the Specialist Group on Coding for Visual Telephony, "Report of the Fifth Meeting in Tokyo (March 25-28 1986)", Contribution #103, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1986.

18. Chairman of the Specialist Group on Coding for Visual Telephony, "Revised Specifications for the Reference Model", Contribution #104, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Jan 1986.

19. Japan, "Proposal of mx64 Kb/s Codec Video Format", Contribution #279, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, May 1986.

20. KDD, NTT, NEC, Fujitsu, "Motion Vector Coding", Contribution #153, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Nov 1986.

21. T. Koga and M. Ohta, "Entropy Coding for a Hybrid Scheme with Motion Compensation in Subprimary Rate Video Transmission," IEEE Trans. Vol.SAC-5, No.7, pp.1168-1174, Aug. 1987.

22. UK, "Differential Motion Vectors 1.", Contribution #200, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1987.

23. Sweden, Norway, Italy, UK, FRG, France, Netherlands, "Motion Compensation Conclusions", Contribution #416, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Dec 1988.

24. UK, Sweden, France, "VLC Set Proposal for nx384Kbits/s", Contribution #122, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Jun 1986.

25. NTT, KDD, NEC, Fujitsu, "VLC Structure", Contribution #152, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Nov 1986.

26. FRG, "Coding of Coefficients with a Two-Dimensional Table", Contribution #170, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Nov 1986.

27. NTT, KDD, NEC, Fujitsu, "Coding of Coefficients Quantization Index with a Two-Dimensional Table", Contribution #188, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1987.

28. FRG, France, GEC, UK, SEPA, "Hybrid Coding at 64Kbits/s and its Compatibility with 384Kbits/s". Contribution #234, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, May 1987.

29. Norway, France, UK, Netherlands, Sweden, Italy, FRG, "1D vs. 2D VLC for Coding Coefficients", Contribution #435, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Dec 1988.

30. FRG, "A Stuffing Codeword for Several Purposes", Contribution #387, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Sep 1988.

31. Japan, "Improvements in Intra Mode", Contribution #407, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Dec 1988.

32. Japan, "Improvements in Intra Mode", Contribution #457, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1989.

33. UK, France, Netherlands, FRG, "Single and Multiple VLCs in Reference Model 4", Contribution #231, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Jun 1987.

34. Japan, "Block Addressing Method for Macro Attribute", Contribution #357, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Sep 1988.

35. FRG, France, Italy, Netherlands, Norway, Sweden, UK, "VLCs for px64kbits/s Initial Compatibility Check", Contribution #449, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1989.

36. Japan, "Signaling Coded Sub-blocks", Contribution #461, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1989.

37. UK, "A Report on the Merits of using Relative Addressing for Blocks Within a Group of Blocks (GOB) Structure", Contribution #146, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Nov 1986.

38. FRG, France, Italy, Norway, Sweden, Netherlands, UK, "VLCs for px64 Kbits/s Initial Compatibility Check", Contribution #449, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1989.

39. NTT, KDD, NEC, Fujitsu, "Adaptive Quantization (for information)", Contribution #149, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Nov 1986.

40. UK, France, FRG, Italy, Norway, Netherlands, Sweden, "H.261 Buffer Specification", Contribution #542, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Nov 1989.

41. FRG, "Quantizer Allocation by Type 3", Contribution #384, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Sep 1988.

42. France, "Regulation at the Macroblock Level", Contribution #480, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1989.

43. Netherlands, FRG, France, "Demonstrations of Hybrid Coding at n * 384 Kbits/s and m * 64 Kbits/s", Contribution #266, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Jan 1986.

44. Chairman of the Specialist Group on Coding for Visual Telephony, "Report of the Tenth Meeting in Red Bank (Oct 13-16 1987)", Contribution #103, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1986.

45. Netherlands, France, FRG, Italy, Sweden, BTRL, "Description of Ref. Model 5 (RM5)", Contribution #339, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1988.

46. FRG, "An Identical Size GOB for Full CIF and Quarter CIF", Contribution #385, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Sep 1988.

47. Chairman, "Transmitted Elements when all Blocks in a GOB are Intra Coded", Contribution #184, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1987.

48. Netherlands, "Post Processing (for information only)", Contribution #207, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1987.

49. FRG, "Motion-Compensating Prediction with Fractional-Pel-Accuracy", Contribution #300, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Jan 1988.

50. Sweden, Norway, Italy, UK, FRG, France, Netherlands, "Motion Compensation Conclusions", Contribution #416, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Dec 1988.

51. Netherlands, France, FRG, UK, Italy, Sweden, Norway, Ireland, Greece, "Pre- In- and Post Codec Filtering", Contribution #475, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1989.

52. Chairman of the Specialist Group on Coding for Visual Telephony, "Report of the Twelfth Meeting in The Hague (Mar 22-25 1988)", Contribution #346, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1986.

53. Japan, "Report on IDCT Mismatch Experiments", Contribution #559, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Nov 1989.

54. UK, "The Requirement for Clipping in the Coding Loop", Contribution #143, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Oct 1986.

55. Chairman of the Specialist Group on Coding for Visual Telephony, "Report of the Sixteenth Meeting in The Hague (Jun 13-16 1989)", Contribution #540, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Jun 1989.

56. S Okubo, R Nicol, B Haskell, S Sabri, "Progress of CCITT Standardization on n x 384 Kbits/s Codec", GLOBECOM 87, pp. 2.2.1-2.1.4, Dec 1987.

57. S Okubo, "Reference Model Methodology – A Tool for the Collaborative Creation of Video Coding Standards", Proc. IEEE, Vol. 83, No. 2, pp. 139-150, Feb 1995.

58. L Stenger, "Standardization of Digital TV Transmission", Proc. Signal Processing of HDTV II, Elsevier, 1990.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

Section V Modern Coding Standards

MPEG2 Tools (Main Profile)

MPEG2 is an extension of MPEG1 in which maximum effort was made to provide compatibility. Absolute backward compatibility does not exist, and would not in fact make any sense – MPEG2 is designed for applications having larger formats and higher bitrates, so it is impossible for pre-existing MPEG1 decoders to receive and decode MPEG2 bitstreams. However MPEG2 decoders are required to decode MPEG1 bitstreams so the standards are forward-compatible. The basic difference between the standards was driven by the targeted applications. MPEG1 was designed for compact disk applications, and was constrained by the disk format to a bitrate of 1.5Mbits/s. This in turn led to a practical maximum picture resolution that was one fourth that of broadcast TV, and that led naturally to a picture format that was progressive scan only. MPEG2's goal was to code standard interlaced broadcast TV at full resolution. Many of the changes between the MPEG2 and MPEG1 normative requirements are described in Annex D of the MPEG2 standard[56] [V-1]. The fine details are not important here, and in summary:

- Many of the changes provided extensions of the MPEG1 standard for such things as larger picture formats, larger motion vector range and larger dynamic range.
- Some changes were made to the data format reflecting the focus of MPEG2 on broadcast TV. In particular, MPEG2 is designed for interlaced coding.
- Some simplifications were made in cases where MPEG1 had attempted backward compatibility with H.261.

The substantive changes to the coding algorithm are summarized below.

MPEG2 additions and modifications to MPEG1:

Coding Tool	Comments
Interlaced prediction modes	
Field-frame DCT	
Alternate scan	Previously studied during H.261 development.
Intra VLCs	Previously studied during MPEG1 development.
Non-uniform quant scale	Previously studied during H.261 development.
Chroma quantization matrices	Separate downloaded chroma matrices were considered during the MPEG1 development.
Intra DC-term precision	
Wider motion vector range	
IDCT Mismatch	Improved, simpler method.
Concealment vectors	

[56] The list in Annex D is not complete and also has an inconsistent format. Thirteen changes are described in detail, and then many more changes are merely listed as syntax elements that can be set to make an MPEG1-compatible bitstream.

The biggest change between MPEG1 and MPEG2 by far was the extension of the MPEG1 algorithm to code interlaced data. There were two components to this change – interlaced motion compensated prediction modes and the DCT format. Interlaced coding was not extensively studied before MPEG2. While for example the Bell Laboratories Picturephone systems employed an interlaced format, the data was simply encoded as individual fields, and the relationship between the fields was ignored. In other words, if the early systems performed motion compensated prediction, they almost always used simple field prediction. There were limited efforts to predict fields of opposite polarity by spatially interpolating the missing lines, with understandably poor results. One reason for the lack of attention to interlaced coding was that most of the research in digital image processing from the 1960s on was performed by computer simulation, and used progressive-scanned images, usually square in size, and typically 512x512 pixels. Naturally the pixels were square. There was essentially a disconnect between the development of digital video processing and the traditional analog broadcast TV world. Interlaced coding was intensively studied for the first time during the development of the MPEG2 standard, both in the competitive phase leading up to the MPEG2 proposals and tests, and within the committee during the collaborative phase as the technology converged to the final standard. Many companies offered proprietary solutions in their proposals for the MPEG2 tests in 1991, with a bewildering array of schemes for predicting fields from frames and vice versa. It is tempting to say that every possible combination was proposed in one way or another! In that sense, it was the job of the committee to sort it all out and reduce it to a small set of effective modes in the standard. A detailed description of the development of interlaced modes is not given here. In summary, the committee standardized a flexible set of rules for field and frame prediction with close to a minimum number of restrictions on the freedom of encoders to dynamically select whichever mode will produce the best results for the data at any given instant. Nevertheless, the experience of increasingly sophisticated encoder technology over the ten years since then has shown that only a subset of the modes are used in practice, and with this hindsight, the standard could be simplified in this area.

The format of the DCT for interlaced coding was however studied very early by Jones in 1978 as discussed earlier [III-39]. The issue is the vertical correlation between lines of pixels in fields and frames. Due to the wider spacing between lines in a field, the correlation is less, and therefore the decorrelating effect of the DCT is lessened. If the lines of the two fields in a frame are interleaved for spatial processing by the DCT, (i.e., if the two fields are processed as a progressive-scan frame) the higher correlation will result in higher coding efficiency. Jones demonstrated this by coding entire frames each way and comparing them. However, while this is globally true for the frame, it is not true for local areas of the frame in which there is significant motion, especially horizontal motion. Objects will be displaced between the fields in such areas by the motion occurring during the field interval, and vertical correlation will be reduced not increased. This will result in lower performance, especially because fast-moving objects with high detail will produce spurious high vertical frequencies in the frame, which will produce significant coefficients in the high-frequency region of the cosine transform domain. Such coefficients will be expensive to code. Efforts to use matched filtering to correct for

this problem produced the inevitable result that the motion was blurred! The problem was addressed by de With 1989 [V-2], who described switching between field and frame formats on a block basis.

A number of coding tool options that had been explored during the H.261 and MPEG1 developments and reduced to a single specification by those standards were opened up in MPEG2 to provide more flexibility for encoders. An alternate scan was introduced that was optimized for field data and provides an alternate approach to the interleaved frame DCT discussed in the last section. This scan pattern partially compensates for the lower vertical resolution in field data. Intra VLC tables were developed for MPEG2 that were specifically optimized for this data type. A non-uniform quantization scale was introduced that provides a wider dynamic range by a non-uniform expansion of the higher amplitude levels. Separate downloadable quantization matrices were provided for chrominance blocks. To accommodate the broadcast TV data type, the option was provided of higher precision for the DC-term.

The range of motion vectors was increased again for MPEG2 and the method of encoding the motion vectors was further developed. The original H.261 scheme merely encoded an integer range of motion vector differences, using a trick to save one bit by exploiting the maximum range of the motion vectors. This scheme was extended by MPEG1 to provide a set of ranges for motion vectors. A dual VLC-FLC mechanism was introduced that provided variable dynamic range supplemented by a fine precision for motion vectors at the extended range. MPEG1 added the option of ½-pixel precision. In MPEG2 the range was extended further. MPEG2 provides ½-pixel precision for all motion vectors.

MPEG2 provides an error concealment mechanism for intra pictures in which macroblocks can be predicted from within the intra picture. The motion vector mechanism is used to signal this. This is the same concept as intra-frame prediction that was explored extensively in the 3D DPCM algorithms.

H.263

To be written.

MPEG4 Tools (Simple and Advanced Simple Profiles)

Development of the MPEG4 standard [V-3] focused on providing additional functionality, rather than improvement of compression performance. The enhancements of video coding were modest and are summarized here. At this time all product development activity for MPEG4 video in the marketplace is based on the Simple and Advanced Simple profiles[57]. Notably these profiles do not include interlaced video coding or bi-directional prediction.

[57] There is significant product development utilizing many other MPEG4 technologies such as audio coding and SNHC (Synthetic-Natural Hybrid Coding).

In the same timeframe as MPEG4 development, H.263 was being developed [V-4]. The two projects maintained a strong liaison, and many coding experts participated in both activities. H.263 focused on further improvement of coding efficiency, and development in this standard preceded MPEG4 in this area. Unsurprisingly there was cross-fertilization and MPEG4 adopted many of the H.263 techniques.

Both standards passed through an evolutionary process, and were released in a series of versions. This paper is based on Version 2 of MPEG4, known officially as Amendment 1, and on Version 2 of H.263 known as H.263+. As such, this draft should be read with caution, because later versions of the standards modified certain parts of the normative requirements.

Video Object Planes – VOPs

A principal goal of MPEG4 was to provide object-based coding. In the context of video coding, which is limited by the source format to a succession of 2D pictures (frames), only a modest step toward this goal could be taken. The video scene can be decomposed into a set of 2D layers of video, each (presumably) containing a semantically meaningful object. Each layer can be coded independently with spatial and temporal resolution pertinent to that layer. The layers can be composited into the complete scene at the decoder. The most typical example comprises a static background, against which one or more foreground objects move dynamically. This concept originated in the formative stages of MPEG4 (1993-1994). One way of looking at what MPEG4 does is simply moving part of the normal post-production process for authoring video from the encoder side to the decoder. In the Simple and Advanced Simple profiles of MPEG4, the VOPs are restricted to rectangular shape, and the video coding process is equivalent to coding a set of video windows of differing sizes and offset locations. This is similar to the Philips CD-I application of MPEG1[58]. Since the Simple and Advanced Simple Profiles do not use bi-directional prediction, only I-VOPs and P-VOPs can be used.

Reduced resolution VOPs

A simple 2:1 reduction in resolution in each dimension can be used to scale macroblocks and reduce the volume of data transmitted. It was introduced in H.263, and is intended for use when there is very high motion.

Deblocking filter.

This tool was also introduced in H.263 and is used only in conjunction with Reduced Resolution VOPs. The borders of blocks are smoothed with a simple one-dimensional filter. This filter is inside the prediction loop.

AC/DC prediction.

[58] Note that MPEG4 Systems can be used to provide normative composition of the VOPs into the scene.

H.263 extended the DC-term prediction used in MPEG1/2 and provided prediction of AC-terms also.

4-MV (four motion vectors); Direct Mode.

This technique is also known as 8x8 motion compensated prediction. Considerable investigation of the use of 8x8 motion vectors was made during the MPEG1 development. At that time it was called "8-escape" or "8-esc". At that time the cost of transmitting four times as many motion vectors equaled or outweighed the coding gain from more accurate motion estimation and the technique was dropped. This seemed to confirm the original motivation for the macroblock concept, which was to share motion vector and other side information between multiple (luminance) blocks. In H.263 and then MPEG4, the technique was reintroduced in a manner that addressed the efficiency of motion vector coding. This was called Direct Mode and was based on the frame interpolation technique[59]. The original approach in frame interpolation was to derive a motion vector for a dropped frame from the motion vector for a subsequent frame. In direct mode, this technique is enhanced by transmitting a delta motion vector. This enables computation of both an accurate forward motion vector for the interpolated frame and a backward motion vector. Therefore the advantages of bi-directional prediction are gained without transmitting two full motion vectors.

Unrestricted MV.

Beginning in H.263, a small change was made to motion vector definition at the boundary of the frame. While MPEG1 and MPEG2 restrict motion vectors to fall within the frame, in H.263 and MPEG4 the motion vectors may point to regions outside the frame. In such cases, reconstruction uses the closest pixel on the boundary of the frame. This technique achieves a coding gain due to improved motion vector prediction.

Motion Vector prediction.

Beginning in H.263, the prediction of motion vectors was enhanced with a two-dimensional prediction from macroblocks above as well as to the left of the current macroblock. The median value of these motion vectors is used as the predictor.

Quantization scale signaling.

Beginning in H.263, a differential quantization scale was signaled at the macroblock layer rather than the whole value. Versions of H.263 and MPEG4 have implemented this in different ways.

Error resilience.

[59] This was called PB-frames in H.263 ver. 1, and was modified to make it work properly in H.263 ver. 2.

H.263 introduced several techniques to improve error resilience that were adopted by MPEG4 with modification in some cases. The slice structure in MPEG1 and MPEG2 already provided a measure of error resilience since the slice header contained the vertical position of the slice. MPEG4 provided additional capability to identify the macroblock address and optionally to download further information necessary to restart decoding if synchronization had been lost[60]. A mechanism called "video packet" was created to facilitate this.

Both standards also included data partitioning for transmitting more critical information including side information in a separate stream from the coefficient information. This separate stream would be lower volume and could be provided with error protection at reasonable cost.

The known technique of reversible VLCs was used by H.263 and adopted by MPEG4. However, while H.263 protected the motion vector data in this manner, MPEG4 applied the technique to the coefficient data.

Background prediction memory – "NEWPRED"

A technique known as NEWPRED was introduced in H.263 in which additional memories for prediction could be allocated at the encoder and receiver, and reference to them could be made in addition to the regular reference memories. One motivation was to provide an error recovery mechanism such that if the normal reference picture were lost, then an alternate reference might be used. This was envisaged to operate in conjunction with signaling in a back channel in a videoconferencing application. However, another interesting use of such a memory is to provide a long-term reference picture that could be used to assist in solving the uncovered background problem. In videoconferencing such a memory could be used to capture the entire scene before any participants entered into it. Such an approach offers a good alternative to bi-directional prediction, because it not only solves the occlusion problem in many cases, it also has the potential to provide the same benefit of noise suppression. As noted earlier, such a concept was not new. For example, H.261 contribution #51 1985 [V-5] describes the use of a background memory that is filled from areas that do not change for a long time[61]. H.263 permitted up to five reference memories to be defined. MPEG4 adopted the same technique.

Dynamic resolution conversion.

This tool is used in conjunction with NEWPRED. It adapts the resolution or if necessary warps the prediction reference to the current frame. In the Simple and

[60] It is also possible in H.263 to signal that GOBs (Groups of Blocks) should be coded independently to facilitate error recovery. In such cases, the GOB boundary would be treated the same as a picture boundary for the purposes of motion estimation.

[61] This contribution described a 3D DPCM algorithm. It was not hybrid coding, but the principle for background prediction is the same.

Advanced Simple profiles, this is restricted to a simple 2:1 resolution reduction in each dimension.

Section VI References

1. ISO/IEC 13818-2, "Generic Coding of Moving Pictures and Associated Audio Information: Video".

2. P H N de With, "Motion Adaptive Intraframe Transform Coding of Video Signals", Philips Journal of Research, Vol. 44, Nos. 2-3, pp. 345-364, 1989.

3. MPEG4 Standard. ISO/IEC 14496-2:1999/Amd.12000(E)[62]. WG11 N3056. Maui Dec 1999.

4. H.263 Standard. "Draft Text of Recommendation H.263 Version 2 ("H.263+") for Decision[63]. Contribution COM 99-E, Study Group 16, ITU-T, 1998.

5. FRG, "Concept of a Predictive Videoconference Codec for Transmission Rates between 64Kbits/s and 2Mbits/s", Contribution #51, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Sep 1985.

[62] File name: W3056.doc. # WG11 N3056. Maui Dec 1999. ISO/IEC 14496-2:1999/Amd.12000(E).
[63] File name: Draft21.doc. COM 99-E. "Draft Text of Recommendation H.263 Version 2 ("H.263+") for Decision.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

Section VI Small Coding Tools

Side Information and Bitstream Formatting

The major algorithm tools – motion-compensated DPCM temporal coding and adaptive block transform spatial coding – provide almost all the compression performance. The small coding tools provide two functions – additional compression algorithms, and formatting utilities for the bitstream. The additional compression algorithms may provide incremental coding features for the main tools or coding for side information. In many cases, a small tool provides both bitstream formatting and a contribution to incremental coding efficiency. This section summarizes all of these tools for the MPEG1 and MPEG2 standards. The layer of syntax used to signal the tool is shown in parenthesis.

Incremental coding features:

Perceptually weighted, downloadable quantization matrices (Sequence).

The ability to more coarsely quantize the higher frequency information was recognized in the 1950s as discussed earlier. (References Kretzmer 1956 [I-4], Schreiber 1959 [I-10]). Wintz suggested perceptually weighting the transform coefficients in 1972 [II-16]. The option to download weighting matrices with user-defined weights was introduced in the JPEG standard. The method is described in the book by Pennebaker and Mitchell 1993 [VI-1]. Lohscheller 1984 [VI-2] is cited as the first use of perceptually weighted coefficients with the DCT. JPEG provides the capability to download separate tables for four components of the signal. Typically only two are used – for luminance and chrominance. MPEG1 adopted this basic approach, but unlike JPEG, MPEG defined separate matrices for intra- and inter-coded data and specified default matrices. MPEG2 extended the MPEG1 approach to use four matrices, with separate matrices for luminance and chrominance.

Full-pixel and ½-pixel prediction (Picture, MPEG1 only)

Prediction with sub-pixel accuracy was considered almost from the very beginning as discussed in the earlier section. See for example Brofferio and Rocca 1977 [III-43]. As noted earlier, Netravali 1979 [III-47] studied accuracies up to 1/128-pixel, and concluded there was no improvement beyond 1/8-pixel. Many researchers in the 1980s included sub-pixel accuracy in their systems. H.261 studied this issue carefully and performed tradeoffs against the loop filter, considering both quality and implementation complexity. At that time, the loop filter was chosen. MPEG1 revisited this issue in 1990, by which time implementation complexity was a lesser concern. MPEG chose sub-pixel accuracy over the loop filter, but provided the option in the bitstream at the picture level to use only integer-pixel prediction. MPEG2 uses only ½-pixel prediction. The concept of switching side information, i.e., attributes, at the picture level as opposed to the block level was introduced in H.261. See for example Jan 1986 contribution #68 [IV-9].

DC-term prediction (MB)

The block size used for compression was chosen to optimize the exploitation of correlation between pixels and the cost of implementation. Residual correlation exists, in particular for the average intensity or color of a region of the image. This is represented by the DC-term of the block transform. Intuitively, regions of typical scenes with uniform color and brightness are much larger than 8x8. This means the DC terms are correlated and can be jointly coded efficiently. The H.261 contribution from FRG, France, GEC, UK and SEPA 1987 [IV-28] that introduced the macroblock concept also introduced the prediction of DC-terms (within the macroblock).

MV prediction (MB)

A similar argument applies to the field of motion vectors. In the model of rigid-body translation assumed for motion-compensated prediction of video data, relatively large objects move with uniform velocity. The prediction block size was chosen to be small enough to capture local movements with sufficient accuracy, particularly on the edge of moving objects. Therefore if the motion estimation is performed carefully, the motion vector field will be uniform within an object. This was exploited by Koga et al in 1983 [III-65] by differentially coded the motion vector information. The subject was investigated thoroughly during the H.261 development, including the technique noted earlier of jointly coding the vertical and horizontal components of the vector with 2D VLCs. See contributions #88 1986 [IV-13], #153 1986 [IV-20], #200 1987 [IV-22].

1st coeff. (Block)

After the introduction of CBP (coded block pattern) in H.261 it was realized that EOB could never be the first coded coefficient in a block. Since the VLC for EOB is very short, a saving was realized by eliminating EOB from the VLC table for the first coefficient and reutilizing this code. As noted earlier, this was attributed to Morrison in 1989 [IV36].

EOB (Block)

The end-of-block code was introduced by Chen in the Scene Adaptive Coder 1981/1984 [II-19][II-20]. It was adopted by H.261, see for example contribution #122 1986 [IV-24].

Side information coding:

Skipped macroblocks, macroblock address, escape codes (MB)

H.261 first introduced skipped blocks in the GOB structure (group-of-blocks, i.e., Slice) with relative addresses. See contribution #85 1986 [IV-12], and subsequent refinements including #122 1986 [IV-24] that included the escape code mechanism.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

Macroblock Type VLCs (MB) #134

Signaling for block attributes was introduced in H.261 before the macroblock concept itself. Block-level change to the quantization scale factor was included in the early proposals for the video multiplex, see contribution #85 1986 [IV-12]. Prediction mode was discussed in contribution #69 1986 [IV-10], and part of the video multiplex proposal in contribution #85. Combining all of these into a single VLC that exploited the joint probability of their events was a feature of the early multiplex contribution #69. Coded block pattern was introduced in contribution #357 [IV-34] together with VLCs for coding the patterns.

MV Range (Picture, MB)

The motion vector range was fixed in the videoconferencing systems and the H.261 standard. In such applications the narrow focus of the application allows such a constraint. MPEG1 was created as a generic standard, so a mechanism was introduced to cope with wide variations of motion range for different applications. This mechanism allowed the range of motion vectors to be set at the picture level. In MPEG2 this mechanism was revised to provide both range and precision for motion at all speeds.

DC-term size (MB)

Various values for the number of bits used to code the DC-term were used by various coding schemes over the years. For example the Scene Adaptive Coder used 9-bits with its 16x16 DCT. H.261 used 8-bits, as did MPEG1. In MPEG2 it was desired to provide the option of higher quality, so a range of bits was provided.

Bitstream formatting:

Prior to H.261, the various manufacturers of videoconferencing equipment created proprietary formats for transmitting and synchronizing their data. Naturally such systems could not interoperate. Contribution #82, 1986 [VI-3], provided the basis for the bitstream format of H.261, which was called the video multiplex. The motivation for this work was the development of a "minimum implementation" that all the laboratories in Europe could use to facilitate interoperability trials. The origin was the field trial equipment these laboratories developed for the H.120 standard. In 1986, the video multiplex was proposed in more detail in contribution #85 [IV-12], revised in contribution #102 [IV-16], and was embodied in the first reference model (RM1) in the 5th meeting report. See Annex 4 of #103 [IV-17]. This provided a basic layered structure and the attribute mechanism that has formed the basis for MPEG syntax ever since. The introduction of normative elements of the bitstream syntax is given below:

Start codes (All layers)

Start codes were used in the early Picturephone systems; see for example 1971 [III-13]. They were introduced in H.120 in the form that included a vertical position code in 1984. They first appeared in H.261 in contribution #82 1986 [VI-3], and were revised to their final form of fifteen zeros and a one plus a five bit

identification code in contribution #147 1986 [VI-4]. MPEG1 extended the code to twenty-three zeros and a one plus an eight-bit identification code.

Closed GOP/Broken Link (GOP)

When the Group-of Pictures layer was first introduced in MPEG1, it was conceived that a group of pictures would be coded as a complete entity that could be decoded without reference to other pictures. It was intended to support random access and editing of the bitstream. This meant that the GOP would end with a P-picture that would be adjacent to the I-picture beginning the next GOP. It was subsequently realized that this was inefficient, and to maximize performance there should be B-pictures between the P-picture and I-picture. This meant however that in general the GOP was no longer independent since the P-picture from one GOP and the I-picture from the other GOP were needed to decode these pictures. Two facilities were provided to give flexible options for this situation. First, it was made possible to make the GOPs independent by not having the B-pictures and signaling that with a flag called "Closed GOP". Second, if the GOP was created with intervening B-pictures, but subsequent editing rendered them un-decodable, this could be signaled with a flag called "Broken Link". It was later decided that these facilities were not needed in practice, and MPEG2 made the entire GOP layer optional.

Picture Type

Signaling of the picture type was introduced in H.261 when intra-pictures were being considered. See contribution #184 1987 [IV-47].

Temporal Reference

The temporal reference existed from the beginning in H.261 (see contribution #82 1986 [VI-3]). It was originally designed in conjunction with temporal subsampling. The decoder could detect when frames were dropped, because the temporal reference of the next transmitted frame would be incremented by the total number of frames. Although this element exists in MPEG, it is irrelevant, because the decoder time stamps convey the same information.

Buffer Delay

In the early video multiplex schemes of H.261, the buffer state was transmitted at the start of each picture. This was changed when the HRD (hypothetical reference decoder) was introduced. The HRD places constraints on encoders that ensure decoder buffers will not overflow. In MPEG1, which begins a sequence with an intra picture necessarily having a large number of bits, a parameter (vbv_delay) was introduced to tell decoders when to begin decoding this first picture. Typically it will take several frame times for the intra picture to be transmitted, so decoding cannot start during that time. The delay parameter was included in all pictures to support random access. In practice, most if not all decoders use the time stamps to control decoding, and the vbv_delay parameter is redundant.

Slices; Vertical Position (Slice)

Coding the vertical position as a part of the start code was first introduced in H.120 for groups of eight lines, and used subsequently in H.261 for the GOB structure. See contribution #85 1986 [IV-12].

Macroblock

The macroblock structure was introduced in H.261 in contribution #234 1987 [IV-30]. The attributes were the same as specified in contribution #141 1986 [VI-5] that describes RM2. A paper describing macroblocks was published by Vogel 1988 [VI-6]

There are also a number of items not related to efficiency, but part of the coding strategy, for example forced update.

Section V References

1. W B Pennebaker, J L Mitchell, "JPEG Still Image Compression Standard", Van Nostrand Reinhold, 1993.

2. H Lohscheller, "Subjectively Adapted Image Communication System", IEEE Trans. on Comm., Vol. COM-32, No. 12, pp. 1316-1322, Dec 1984.

3. UK, FRG, France, Italy, Netherlands, Sweden, "Specification for the Flexible Prototype 2nd Generation Videoconference Codec", Contribution #82, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Mar 1986.

4. UK, "Bit Pattern for PSC and GBSC", Contribution #147, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Nov 1986.

5. Chairman of the Specialists Group on Coding for Visual Telephony, "Specifications for Reference Model Version 2 (RM2)", Contribution #141, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, Sep 1986.

6. V P Vogel, "Videobild auf der Datenleitung", Funkschau, No. 26, pp. 60-63, 1988. (In German)

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

Section VII Performance Contributions

This section provides a guide to the performance provided by each of the coding tools to the overall performance of the standard. Any review of candidate performance measures quickly establishes there is no precise quantitative measure. The earliest papers documented that correlation between measures such as mean square error (MSE) or signal to noise ratio (SNR) and perceived error are poor. During the development of the MPEG standards, formal subjective testing was used to select the basic algorithm for each standard, and again for verification of the refined algorithm. Such testing is time-consuming and expensive, but it is the only reliable method to produce meaningful results. During the development of the standard, as coding tools were being considered for inclusion in the standard, informal viewing tests were used as a practical compromise between time and cost constraints and the quality of the testing. Decisions based on these tests were made by consensus. A rough guide to the usefulness of a coding tool did use SNR as a basis – if the SNR was improved by use of a tool by 0.3dB, it was seriously considered.

A key part of the problem of measuring performance is the selective nature of human visual perception. A classic example is to consider the errors introduced by changing the grey level of the entire picture by a value of one, versus adding a block artifact to the eye of a person in the scene. The MSE or SNR of the former change will be large, while negligible for the latter change. The perceived error however will be very large for the latter change, but the former error will not even be detected. There are two aspects to this. First, errors found to be perceptually critical can occur in a very small percentage of the scene. Notably they occur on edges that are used by the human vision system as cues for information content. Such errors will be statistically negligible when averaged over the whole scene. Second, the perceptually critical errors can be small in magnitude, but involve spatial relationships that are not naturally occurring visual features. This issue is especially important for errors introduced by transform coding, because they tend to be block-based. The whole problem of measuring perceived error is made considerably more complicated when the time domain is included. For example, the earliest analog compression work on pixel interlace seemed fine on paper or when studying a single video frame. But when real-time systems were built, dot patterns were highly visible.

During the development of the standards, tools were added incrementally that provided additional quality while keeping the target bitrate fixed. This did not guarantee that the final set of tools maximized quality at that bitrate, but it is reasonable to argue that it did do so for several reasons. First, that variation in the data itself meant the sensitivity of performance to any of the underlying factors was low, and the optimization was soft. Second, that many alternatives were tried in the process of selecting tools. Third, many independent tests were performed that explored at least the local optimization in each case. It is probable that the MPEG standards have provided the highest quality for generic video coding at the time of completion. Two factors have been discussed earlier that

impact this. First, compromises were made for implementation complexity, and removal of such constraints could result in higher performance. Second, an insistence on backward compatibility meant there was a legacy of coding tools or syntax construction dating back to H.261. In particular, tools introduced for very low bitrate coding, and the specific characteristics of videoconferencing content may not be appropriate for high-quality broadcast TV applications with MPEG2.

Since the MPEG1 and MPEG2 standards were developed, considerable experience has been gained with smart encoding techniques. Many tricks have been developed to minimize artifacts and to maximize perceived quality for the decoded image. Some of these techniques have recognized the effective filter characteristics of the coding process, and the expected statistics of the VLC tables, and have preprocessed the data such that its characteristics and statistics match those expected by the standard. Therefore the coding tools work more efficiently, and the available bits are used optimally to produce the highest quality. Quantitative measure of preprocessor and smart encoder contributions to the performance is not possible, but it is very likely that they at least equal any contributions from most small coding tools or fine-tuning of the coding tools.

In this section, performance will be reviewed in the context of bits/pixel. This is the only meaningful way to indicate the percentage contribution to overall performance from a given coding tool. It is necessary to make a set of assumptions about the data and coding process. First, an alternative to using the tool must be made as a basis for comparison. Unfortunately this can be arbitrary and many alternatives, more or less ingenious could be postulated in each case. Second, interaction between tools means that changing one affects others, and the global optimization is soft in this sense. Third, statistics must be assumed for the adaptive behavior of the algorithm, which further will be dependent on the application, format and bitrate. Therefore, this discussion is an illustration of the performance, not a definitive statement.

Three cases are considered for H.261, MPEG1 and MPEG2 coding using typical picture formats and bitrates. Constant bitrate coding (CBR) is assumed. The basic parameters are shown below. Numerical values have been rounded for simplicity.

The first table provides the budget of bits for each macroblock and block in the three formats, and the average bitrate per pixel. In the ensuing calculations, these budgets may be modified according to the number of skipped macroblocks, not-coded blocks, and macroblock types per picture type.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

Format	#MB	#Blocks	Bitrate	#Bits/MB	#Bits/block
QCIF/10fps	99*10	594*10	64Kbits/s	64.6	10.8
SIF[64]/30fps	330*30	1980*30	1.15Mbits/s	116.2	19.4
601[65]/30fps	1320*30	7920*30	4Mbits/s	101	16.8

Format	#MB	#Blocks	Bitrate	#Bits/pixel[66]	#Bits/pixel[67]
QCIF/10fps	99*10	594*10	64Kbits/s	0.25	0.006
SIF/30fps	330*30	1980*30	1.15Mbits/s	0.45	0.11
601/30fps	1320*30	7920*30	4Mbits/s	0.39	0.39

The second table provides assumptions for the number of bits in each picture type. For MPEG1 and MPEG2, a ½-sec GOP is assumed with two B-pictures between each pair of reference pictures. The GOP therefore contains one I- picture, four P- pictures, and ten B-pictures.

Format	#Bits/I picture	#Bits/P picture	#Bits/B picture
QCIF (64Kbits/s)		6,400	
SIF (1.1Mbits/s)	150,000	50,000	20,000
601 (4.4Mbits/s)	600,000	200,000	80,000

The third table provides assumptions for the number of skipped macroblocks and coded blocks in motion compensated pictures, and for the number of intra-coded macroblocks in P-pictures and B-pictures and the number of forward-only or backward-only coded macroblocks in B-pictures. (Denoted as P MB).

Format	#Skipped MB	#Coded Blocks	#I MB	#P MB
P-frames	1/3	5/6	3/64	
B-frames	2/3	1/2	3/64	9/32

The contribution to coding performance for each stage of processing and each coding tool is given below. The baseline for analysis is the display format of broadcast TV with standard-definition (NTSC/PAL) resolution:

24 bits/pixel 4:4:4 format; 720x480x30/720x576x25 resolution.

[64] NTSC SIF.
[65] NTSC 601.
[66] Rate per pixel, per preprocessed format: QCIF, SIF or 601 resp.
[67] Rate per pixel, standard definition broadcast TV resolution.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

Preprocessing
Considerable compression is effected before any coding tools are used:

- MPEG2 4:2:0 –
  - 704x480 = 337,920; x24bits = 8,110,080 bits
  - 704x480x8 + 2x352x240x8 = 4,055,040 bits
  - 12 bits/pixel
- MPEG1 SIF 4:2:0
  - 3 bits/pixel (352x240x8 +2x176x120x8)
- QCIF/10fps
  - 0.25 bits/pixel

Major tools
The use of transform coding to spatially compress individual frames can produce "good" quality at around 1.0 bits/pixel. Some examples are cited below:

- Scene Adaptive Coder – 0.4bits/pixel, 1984 [III-20]. 512x512x15frames/s. At 30 frames/s this rate would be 0.8bits/pixel. Chen noted visible errors at this rate.
- MPEG1 Intra-coded frames – 0.44bits/pixel. This rate produces the "VCR quality" that MPEG1 targeted. There is an obvious loss of resolution compared with broadcast TV quality.
- MPEG2 Intra-coded frames – 1.56bits/pixel. At this rate, MPEG2 quality is nearly transparent.
- JPEG – 1.2bits/pixel. This rate is typical for images that do not display visible artifacts.

Hybrid coding with motion compensated prediction at ½-pixel accuracy can produce "good" quality at around 0.35 bits/pixel.

- Ericsson – 0.2bits/pixel, 1985 [III-73]. 248x288x25 interlaced frames/s. Ericsson noted visible artifacts at this rate.
- MPEG2 formal tests were conducted at 4Mbits/s (0.39bits/pixel) and 9Mbits/s (0.88bits/pixel) with the expectation that artifacts would be visible at the former rate (allowing comparison between different proposals), while the latter rate would provide transparent quality.

Minor tools
This section estimates the contribution by each minor tool to the total performance. Examples are derived mostly from MPEG1 syntax and semantics. In most cases the procedure and the VLC tables are identical to H.261.

Perceptually weighted, downloadable quantization matrices (Sequence).

The gain from use of the matrices is by definition perceptual and hard to quantify. Tasto and Wintz [II-10] provided very early data, but it is hard to interpret, and is very old. The JPEG book by Pennebaker et al [V-1] provides discussion in terms of visibility thresholds for DCT terms at different frequencies and provides empirically derived tables. However the authors recommend dividing the values in the tables by two in order to produce images indistinguishable from the original. In such a case the impact would be weak compared with no scaling, since the highest weight is then only a factor of four for the highest frequency terms compared to the low frequency terms.

DC-term prediction (MB)

This is used only for Intra-coded blocks. DC prediction can be used within an Intra macroblock for the luminance component, and between successive Intra-coded macroblocks for both luminance and chrominance components. In this analysis, prediction between Intra macroblocks in P- and B-pictures is ignored, since adjacency of such macroblocks is not known. The DC-term is 8-bits while the entropy of the difference is 2.7 bits (based on the MPEG1 VLC Table B.5 for DC term size). The saving is 5.3 bits per coded block. The table below summarizes the savings. No account is taken of resetting the predictors inside the picture, so the figures are optimistic.

Format	#Bits saved per I-frame	#Bits saved per P-frame	#Bits saved per B-frame	Bitrate saving
SIF/30fps	10,494	398	398	2.8%
601/30fps	42,294	1643	1643	3.3%

MV prediction (MB)

Motion vectors are predicted between macroblocks and the difference is VLC encoded. The entropy of the difference is 2.5 bits (based on the MPEG1 VLC table B.4[68], which is the same as the H.261 table). The VLC table provides values over a range of +/-16 integer pixels. This provided a complete solution for H.261, and was sufficient for the motion in typical videoconferencing scenes. In MPEG1 and further in MPEG2, the range was extended. This is discussed later in this section. The H.261 process included a trick to save one bit per motion vector component. Since the motion vector range is limited, a differential motion vector whose magnitude exceeds the range when added to the previous motion vector can be interpreted as wrapping around that range. The savings are thus 2.5 bits per motion vector component, or a total of 5 bits per non-skipped macroblock for P-pictures and 10 bits for B-pictures. The savings are summarized in the table below.

[68] This table is the same as the one for MBA.

Cliff Reader, Ph.D.
P.O. Box 2666, Saratoga, CA., 95070
Ph. 408 741 1698, Cell 408 857 0209
Cliff @reader.com

Format	#Bits saved per I-frame	#Bits saved per P-frame	#Bits saved per B-frame	Bitrate saving
QCIF/10fps		330		5%
SIF/30fps		1100	1100	2.7%
601/30fps		4400	4400	3.1%

1st coeff. (Block)

The 1st coefficient "trick" saves 1 bit per coded block.

Format	#Bits saved per I-frame	#Bits saved per P-frame	#Bits saved per B-frame	Bitrate saving
QCIF/10fps		330		5%
SIF/30fps	1980	1100	330	0.3%
601/30fps	7920	4400	1320	0.34%

EOB (Block)

The EOB VLC code is very short – only 2-bits. If it were not used, and no alternative mechanism were provided, many bits would be needed to signal the final run of zeros with a VLC. An alternative would be terminating using a FLC – 63 zeros is the maximum so 6 bits would suffice, but the FLC would have to be signaled with a special code that in itself would have the same effect as EOB. Another alternative would be to use a separate VLC table for "last coefficient". Optimizing a different set of VLCs would probably reduce efficiency by 2-3 bits per coded block, so the bitrate saving of EOB can be estimated to be around 1%.

Skipped macroblocks, macroblock address - MBA (MB)

If macroblocks were not skipped, then MBA would not be needed. Coding MBA costs an average of 2.5 bits based on the MPEG1 VLC Table B.1. Alternatives would be a not-coded MB type, or CBP with all-not-coded. These alternatives might add 1 bit to the VLC tables, so there would be a net saving of 1.5 bits. Note the H.263 standard uses a simple 1-bit flag for not-coded macroblocks, which is another alternative. The following table shows the cost (negative saving) in bitrate of using skipped macroblocks with macroblock address.

Format	#Bits saved per I-frame	#Bits saved per P-frame	#Bits saved per B-frame	Bitrate saving
QCIF/10fps		<50>		<0.78%>
SIF/30fps		<165>	<330>	<0.77%>
601/30fps		<660>	<1320>	<0.89%>

Macroblock Type VLCs (MB)

The macroblock type VLC signals the existence of five entities in the bitstream – macroblock quantization scale factor (Mquant), forward motion vector, backward

motion vector, intra-coded macroblock and coded block pattern (CBP). From these, the prediction mode of the macroblock and whether it is coded or not-coded can be deduced. The VLC has an average length of 2.8 bits based on the MPEG1 VLC table B.2. If MB type were not used, the following FLCs would result:

- Mquant flag 1-bit
- Pred. Mode 2-bits
- CBP flag 1-bit

So macroblock type provides a net saving of 1.2 bits per non-skipped macroblock, and is summarized in the table below. Note that after the introduction of macroblocks in H.261, contribution #266 [IV-43] compared their performance against the prior method for video multiplex. It reported a coding gain for 64Kbits/s but no gain at 384Kbits/s.

Format	#Bits saved per I-frame	#Bits saved per P-frame	#Bits saved per B-frame	Bitrate saving
QCIF/10fps		40		0.63%
SIF/30fps		132	264	0.62%
601/30fps		528	1056	0.71%

One motivation for macroblocks was sharing of attributes between blocks, thereby reducing side information. Since the original introduction, there have been advances in coding that change the homogeneity of blocks within a macroblock. For example prediction, with 8x8 motion vectors may be used.

Coded Block Pattern – CBP (MB)

Coded block pattern was introduced as an alternative to using EOB in all blocks, including those with no coefficients. Therefore it replaced 2 bits in each not-coded block in non-skipped macroblocks. The entropy of the VLC table is 5 bits (based on the MPEG1 Table B.3). The saving is summarized in the table below.

Format	#Bits saved per I-frame	#Bits saved per P-frame	#Bits saved per B-frame	Bitrate saving
QCIF/10fps		<33>		<0.52%>
SIF/30fps		<110>	385	0.59%
601/30fps		<440>	1540	0.68%

MV Range (Picture, MB)

Motion vectors in MPEG1 and MPEG2 are scaled by a factor provided at the picture level, and if the range exceeds that of the VLC table, (+/-16) a residual value is also transmitted as a FLC. The length of the FLC is given by a range factor transmitted at the picture level. This scheme was developed to preserve some measure of backward compatibility with H.261. The alternative would be to transmit a simple VLC covering the complete motion vector range, including the usual escape code mechanism to limit the VLC codes to reasonable length. It is

not possible to quantify the MV range process without statistics on motion vector differences across the whole range.

Closed GOP/Broken Link (GOP)

As discussed earlier, when the GOP layer was originally introduced, it was conceived as a mechanism to allow access into the bitstream and to facilitate editing. As such it was a self-contained entity that could be decoded without reference to data outside the GOP. This meant it must terminate at each end with I- and P-pictures. Such an entity is called a closed GOP. However, this resulted in a bitstream in which GOP boundaries contained adjacent P- and I-pictures. This was recognized as inefficient because the temporal correlation between them was not exploited. The flexibility was introduced to code B-pictures between them, resulting in an entity called an open GOP. If, however the bitstream were edited, these B-pictures could not be decoded if one of the I- or P-picture predictors were removed by the edit, and this may be signaled by the broken link flag. This flag does not impact performance. If the bitstream is coded with only closed GOPs, the performance impact is hard to quantify, because if the bitrate is fixed, the effect is to replace one B-picture with a P-picture. As an example, consider MPEG1 with a ½-sec GOP with and without an end B-picture to show reallocation of bits. If the number of bits for I- and P-pictures remains the same as before at 150K and 50K, the number of bits per B-picture declines from 20K to 16.67K.

Summary

The origin and development of coding tools used in the MPEG standard has been traced from the pre-digital era up to Version 2 of the MPEG4 Standard. In the early digital period, the focus in any of a number of research institutions was to develop a coding solution based on a single mathematical technique. Two fundamental strategies were pursued – DPCM and transform coding. The single approach was found to be sub-optimum, and hybrid coding was established by the end of the 1970s as the basis for efficient, practical coding. DPCM became the basis for temporal coding, while transforms became the basis for spatial coding. In the early period it was also established that adaptive coding was essential. Attempts to provide a mathematical model for characterization of the data and hence an analytical coding solution were doomed by the non-stationary nature of the image or video data statistics. Even then, while some non-stationarity could be modeled and tracked during the coding process as parametric variation of a stochastic model, discontinuities in the data meant that the models were not even valid when spanning such a discontinuity. This situation led to a period of experimentation and refinement, with steady, incremental improvement of performance as experience with adaptive tools was gained. It was found that mathematical optimization was not effective, due to weak relationships between minimum performance of the model and the parameters driving it. Any benefit of a mathematical solution was also muted by the strong influence of psychovisual factors. Coding gains by exploiting

the inability of the human vision system to perceive spatial and temporal resolution are significant and have been key in achieving the composite performance of the MPEG algorithms.

Over the entire period, including the present time, the cost of implementation has been a powerful influence on the algorithm design. In the early period, this was often an overriding factor, with coding efficiency or reconstructed quality being sacrificed for practical implementation. Despite this, the cost was too high for commercial use for many years. The sheer volume of video data meant that both the cost of computational logic and the cost of processor memory were prohibitive. Many early developments were forgotten, only to be reinvented in the 1990s when implementation became practical. Today, further revision is taking place as techniques rejected during early H.261 and MPEG development for cost reasons now seem reasonable to implement, even in mass-production consumer products.

While the early developments took place in individual institutions, in the 1980s it became more and more the practice for development to take place in a collaborative environment. In Europe this was fostered by European Community policies and programs. Such practice became truly international in the H.261 program, which laid the foundation for modern collaborative working, and was adopted by MPEG with great success. It has since been proven by both positive and negative examples to produce a superior result to any that a single institution can produce. This is probably due both to the sheer volume of experimentation that can be accomplished by multiple organizations and to the open-mindedness that results when the convictions of any one individual or organization are challenged by the differing convictions of others.

Such a process has required very strong management. The common belief that design-by-committee is a recipe for disaster has been denied first by an immutable adherence to schedule. More subtly, the MPEG development process has continuously striven to standardize the absolute minimum technology necessary to achieve the goal of interoperability. Particularly during the MPEG1 and MPEG2 period, the number of small coding tools was parsed to the barest minimum. For the tools that were included, the most generic capability was provided. This has left the encoder the maximum freedom to find the optimum performance point, yet maintaining the assurance that all conforming decoders will be able to function. It provides the flexibility to optimize to specific data types or operational characteristics. Nevertheless, with all the experience gained over the past ten years, it is now possible to see that certain tools could be removed to simplify the standards without any performance impact. At the same time, some of the tradeoffs made in favor of lower cost implementation could be revisited.

The set of coding tools can be divided into classes for the purposes of understanding their contribution to overall performance. Pre-processing is not a part of the MPEG coding process, yet it achieves considerable compression. The major coding tools implement the bulk of compression. A sub-class of the minor coding tools specifically performs additional compression, particularly for side information related to the major coding

tools. The other minor coding tools are related to formatting of the bitstream for transmission or storage and do not contribute to compression per se. However they may provide yet further compression as a side benefit. While the major coding tools are essential for providing the performance of MPEG, the specific minor coding tools are less significant. Alternatives exist that could provide the same performance. This is especially true for the minor tools concerned only with construction of the bitstream syntax. What is important is specifying a particular construction to facilitate interoperability.

The coding tools were invented many years ago in most cases, yet have only found their way into the standards in the past ten years. In this context, the role of the standards organizations has been to provide a forum within which the tradeoffs and experiments could be performed. The distillation of many candidate tools and the parametric optimization of those tools has been a multi-dimensional task requiring thousands of man-years of effort. Only with such a concentration of resources, and the management rigor of the ITU and MPEG committees has this effort been possible.

Reviewers:

- Gisle Bjontegaard
- Wen Chen
- Leonardo Chiariglione
- Stafan Ericsson
- Chad Fogg
- Toshio Koga
- Geoff Morrison
- Sakai Okubo
- Istvan Sebestyen
- Gary Sullivan